

Jan Bosas

MULTIMEDIA  
OPISKELUMATERIAALIN  
TUOTANNOSSA

## VAASAN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma

**TIIVISTELMÄ**

Tekijä	Jan Bosas
Opinnäytetyön nimi	Multimedia opiskelumateriaalin tuotannossa
Vuosi	2010
Kieli	suomi
Sivumäärä	61 + 2 liitettä
Ohjaaja	Päivi Sampola

---

Tämän opinnäytetyö käsittelee multimedian mahdollisuuksia etäopiskelumateriaalin tuottamisessa. Toimeksiantona oli tehdä vähintään kymmenen flash ActionScript 3 -ohjelmointiin liittyvää tehtävää opetusvideoiden muodossa. Tehtävistä koostetaan verkko-opetusmateriaalia virtuaaliseen oppimisympäristöön Moodleen.

Teorian ja empirian avulla oli tarkoitus selvittää, millaista on laadukas ja käyttäjäystävällinen multimedia, sekä luoda sellaista kurssia varten. Lopputuloksena syntyi Moodle-sivusto, 12 tehtävää videoineen, sekä tarvittava ohjeistus ja materiaali tehtävien tekemiseen. Työn lopputulos on tavoitteiden mukainen, joskin lopputuloksen lopullinen arviointi onnistuu vasta, kun materiaali otetaan käyttöön.

---

Asiasanat: Multimedia, käytettävyys, etäopiskelu

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Tietojenkäsittelyn koulutusohjelma

## ABSTRACT

Author	Jan Bosas
Title	Multimedia in the Production of Study Material
Year	2010
Language	Finnish
Pages	61 + 2 Appendices
Name of Supervisor	Päivi Sampola

---

The topic of this thesis is the prospects of using multimedia in the production of online study material. The assignment is to do at least ten Flash ActionScript 3 – related assignments, in the form of video tutorials. These assignments will be composed into online study material that will be put into a virtual learning environment, Moodle.

The possibilities of multimedia in the production of online study material will be studied through the theoretical and practical parts of the thesis. The practical part will also include the making of such material. The end product of the practical work will be a Moodle web page, 12 assignments with video tutorials and the instructions and material necessary for making these assignments. The end result is compliant with the set objectives, even though the final evaluation can only be done when the material is taken into use as study material.

---

Keywords	Multimedia, usability, online studying, self-study
----------	--

1	JOHDANTO .....	4
1.1	Tehtävänanto ja rajausta.....	4
1.2	Toimintasuunnitelma.....	5
2	MULTIMEDIA JA SEN MAHDOLLISUUDET .....	6
2.1	Multimedian luonne .....	6
2.2	Multimedian eri muodot.....	7
2.2.1	Typografia .....	7
2.2.2	Kuva.....	8
2.2.3	Ääni.....	10
2.2.4	Video.....	13
2.3	Väylät .....	16
2.4	Ohjelmistot ja formaatit.....	19
2.5	Multimediatuotteen käytettävyys.....	22
2.6	Multimedia opetusmateriaalin tuottamisessa.....	25
3	OPETUSMATERIAALIN TUOTTAMINEN.....	28
3.1	Tehtävien toteutus .....	28
3.2	Videoiden suunnittelu ja toteutus.....	28
3.3	Opetusmateriaalin jakaminen verkossa.....	30
4	TUOTETTU VERKKO-OPETUSMATERIAALI.....	31
5	YHTEENVETO .....	37
5.1	Prosessin kuvaus ja arviointi .....	37
5.2	Tulokset.....	41
5.3	Tulosten hyödyntäminen ja jatkotutkimus .....	43
	LÄHTEET .....	44
	LIITTEET.....	45

# 1 JOHDANTO

Tässä luvussa esittelen toimeksiannon ja tehtävänrajauksen. Myös toimintasuunnitelma esitellään.

## 1.1 Tehtävänanto ja raja

Opinnäytetyön aiheen sain Vaasan ammattikorkeakoulun tietojenkäsittelyn osastolta. Toimeksiannon tarkoituksena on multimediaopetuksen monimuotoistaminen. Tehtäväni oli tehdä ainakin kymmenen Flash ActionScript 3 (AS3)-opetusvideota itsenäistä etäopiskelua varten. Nämä opetusvideot muodostaisivat kokonaisuuden, jonka avulla opiskelija voisi opetella AS3-ohjelmointia Moodles-ta itsenäisesti.

Opinnäytetyön tarkoituksena on tehdä esimerkin tapaan toimivia opetusvideoita. opetusvideot koostuvat ruutukaapatuista videoista, jossa teen itse kyseisen tehtävän. Se mitä ruudulla tapahtuu selitetään jälkiäänitetyn selostuksen avulla. Videoiden tueksi oli tarkoitus tarjota tekstiä, kuvia ja kaikki ne tarvittavat tiedostot, joita tehtävien tekemiseen tarvitaan. Esimerkiksi jos tehtävänä on tehdä palapeli, löytyvät palapelin palat kuvatiedostoina Moodlesta. Mitään tarkkaa rajausta sen suhteen ei ollut, mitä tehtävien tulisi pitää sisällään. Kuitenkin niiden tulisi olla monipuolisia, eli kunkin tehtävän tulisi käsitellä jotain uutta asiaa AS3-ohjelmoinnista. Myös vähintään yksi peli tulisi olla mukana opiskelijoiden mielenkiinnon ylläpitämiseksi.

Dokumentin teoreettisen osan tulisi käsitellä multimediaa yleisesti ja vastata tutkimuskysymyksiin. Asetetut tutkimuskysymykset käsittelevät multimediaa. Käytännön työn tavoitteena on opettaa ActionScript 3-ohjelmointia mahdollisimman hyvin, eli niin, että opiskelu on tehokasta, materiaalin käyttö helppoa ja luontevaa, sekä niin, että tarkoitetut asiat tullaan käytyä läpi ja opiskelijat omaksuvat ne mahdollisimman hyvin. Työni on tarkoitus vastata seuraaviin tutkimuskysymyksiin: Millainen on käytettävyydeltään onnistunut multimediatuote? Multimediamahdollisuudet opiskelumateriaalin tuottamisessa? Miten rakennetaan hyvä multimediatuote?

## 1.2 Toimintasuunnitelma

Aloitan kirjallisuuden etsimisellä. Pääpainopisteenä on multimedia ja Flash ohjelmointi. Multimedia kirjallisuudesta saisin tarvittavaa tietoa ja taustaa dokumentin teoriaosiota varten, sekä se helpottaisi minua videoiden ja muun multimedia sisällön suunnittelussa ja toteutuksessa. Flashista taas tarvitsisin teorialtietoa. Käytännön apua ohjelmointiin löytyy erittäin hyvin internetistä mm. Adobelta itseltään ja monilta muilta forumeilta ja sivustoilta. Etsin käsiini myös opinnäytetöitä koulujen kirjastoista ja muutaman sähköisen Internet-tietokannasta. Vertailin niitä keskenään ja tutkin, mikä olisi mielestäni paras tapa toteuttaa oma työni. Toki tässä auttoi paljon myös opinnäytetyöopas.

Seuraavaksi vuorossa olisi opetusvideoiden sisällön suunnittelu. Ideoita tähän sain rutkasti Internetistä sekä aikaisemmin tekemistäni ja näkemistäni Flash-sovelluksista. Kun sisältö oli suunniteltu pääpiirteissään, aloin tehdä tehtäviä Flashilla. Sitä mukaa, kun tein tehtäviä, tarkensin suunnitelmaa loppujen tehtävien osalta.

Kun tehtävät olivat valmiit, aloitin itse videoiden tekemisen. Ne tekisin niin, että tallentaisin tietokoneeni näytöllä näkyvän kuvan, eli työpöytäni sitä varten tarkoitettulla ohjelmistolla. Videoiden sisältö koostuu siitä, että teen tehtävät uudestaan, tällä kertaa tekoprosessin taltioiden. Tämän jälkeen tuli jälkiäänitysvaihe, jossa luon videon ääniraidan. Tarkoitus oli selittää, että mitä tehdään ja miksi. Koodauksen osalta tärkeää oli selittää miksi tehdään se koodi, mitä tehdään ja mitä se tekee ja saa aikaan sovelluksessa. Valmiit videot vein YouTubeen ja linkin Moodleen.

Käytännöntyön viimeistely koostui Moodle-sivuston viimeistelystä, eli tekstien, kuvien ja tarvittavien tiedostojen lisäämisestä. Oli tärkeää ottaa huomioon sivun käytettävyys, etteivät tehtävät olisi sekavia ja tekstit aseteltu miten sattuu. Tästä syystä päädyin jakamaan tehtävät selviksi osioikseen, jotka noudattavat samaa rakennetta ja järjestystä. YouTube-linkit ja koodi aukeavat uudessa ikkunassa, käytön helpottamiseksi ja selkeyttämiseksi.

## 2 MULTIMEDIA JA SEN MAHDOLLISUUDET

Tämä luku vastaa kysymykseen siitä, mitä multimedia on. Myös multimedian eri osa-alueet ja jakelukeinot käydään läpi. Multimedian tuotannossa tarpeellisia ohjelmistoja ja niiden valintakriteerejä pohditaan yleisesti, pääpainon ollessa kuitenkin Flashissä. Lopussa käsitellään vielä multimedian käytettävyyttä sekä soveltuvuutta opetuskäytössä.

### 2.1 Multimedian luonne

Multimedialle on vaikeaa antaa tarkkaa määritelmää ja jokaisella onkin oma näkemyksensä siitä, mitä se tarkoittaa. Perinteinen käsitys on se, että multimediksi voidaan kutsua sellaista informaation välittämistä, joka koostuu jonkinlaisesta yhdistelmästä ääntä, kuvaa, liikkuvaa kuvaa ja tekstiä.

Multimedia voidaan myös määritellä vaikka seuraavasti: ”Multimedia on informaation monikanavaista välittämistä ihmiseltä tai järjestelmältä ihmiselle tai järjestelmälle sekä välitetyn tiedon interaktiivista käyttämistä ihmisen toiminnassa” (Ketamo, Multisilta, 2004, 12). Kyse ei siis ole välttämättä pelkästään ihmiselle suunnatusta viestinnästä, vaan myös koneet voivat kommunikoida keskenään.

Kirja, jossa käytetään tekstin lisäksi kuvia, voidaan luokitella multimediksi. Konsertti, jossa on esiintyjän lisäksi videotaulut, on multimediaa. Yleisimmin multimedia tehdään kuitenkin tietokoneille ja muille tietojenkäsittelyvälineille, kuten DVD-soittimille ja matkapuhelimille. Nämä laitteen mahdollistavat multimedian tehokkaan käytön, niin äänen, kuvan, kuin videonkin. Tietoverkkojen, kuten internetin sekä laitteiden tallennuskapasiteetin, liikuteltavuuden ja levinneisyyden ansiosta, käyttäjä voi monesti valita milloin ja missä hän käyttää multimediaa.

## 2.2 Multimedian eri muodot

### 2.2.1 Typografia

Kirjoitettu teksti on ollut olemassa kauan ennen muita nykyisin yleisesti multimediksi mielletäviä elementtejä, kuten tallennettua ääntä, kuvia tai videoita. Tekstin ensisijainen tarkoitus multimedias-  
sa on välittää tietoa. Tekstillä on useita etuja perusviestinnässä ja informaation tarjoamisessa. Tekstiä voidaan käyttää kaikissa kuviteltavissa olevissa medioissa, se toimii yleensä kaikissa laitteissa ilman tiettyjä päivityksiä tai erityisohjelmistoja. Tekstistä koostuvat tiedostot ovat pieniä ja hidaskin yhteys riittää niiden nopeaan siirtämiseen. (Keränen, Lamberg & Penttinen 2003, 52.)

Kirjoitustyyli ja -tapa vaikuttavat olennaisesti siihen, kuinka uskottavaa teksti on ja siihen, millaisen mielikuvan lukija saa aiheesta ja kirjoittajasta. Myös kirjasimella, tekstin koolla, värillä ja asettelulla on merkitystä niin mielikuvien, kuin luettavuuden ja selkeydenkin kanssa. Olennaista tekstin kirjoitustyylin valinnassa on huomioida kohderyhmä. Nuorille ja eläkeläisille tulee kirjoittaa eri tavalla. Uskottavuus saattaa jäädä vain haaveeksi, jos asiatekstissä on käytetty kirjasimena esimerkiksi *Bradley Hand ITC*:tä. (Keränen ym. 2003, 52, 54)

Käytettäessä tekstiä multimedias-  
sa tulee huomata joitain eroja paperilta luettavaan tekstiin. Tekstiä tulisi olla vähän ja sen tulisi olla selkeää ja ytimekästä. Kappaleiden tulisi olla lyhyitä ja otsikoinnin riittävää ja selkeätä. Yleensä lukija silmäilee otsikot läpi ennen lukemista. Otsikoiden tulisi siis kertoa tiivistetyssä muodossa seuraavan tekstin sisältö. Myös luetteloita, kaavioita ja kuvia olisi hyvä käyttää pelkän tekstin sijaan. (Keränen ym. 2003, 56.)

Mitä vähemmän käyttäjän tarvitsee selata näkymää alas ja ylös, sitä helpompaa on hahmottaa sisältö. Liian suuri tekstin määrä ilman selvää jaottelua ja otsikoita saa helposti käyttäjän kyllästymään. Esimerkiksi jos internetsivuston pääsivulla on heti kaksi ruudun mittaa tekstiä, eikä kuvia, videoita tai selkeitä linkkejä johonkin järkevämpään sisältöön, on hyvin mahdollista, että käyttäjä lähtee sivustolta hyvinkin nopeasti. (Keränen ym. 2003, 56.)



### 2.2.2 Kuva

Ensimmäinen valokuva otettiin vuonna 1826 (Ketamo ym, 2004, 26). Sitä ennen kuvia tehtiin piirtämällä, maalaamalla tai kaivertamalla. Vuonna 1975 otettiin ensimmäinen digitaalinen valokuva. Digitaalisessa muodossa olevat kuvat ovat olennainen osa multimediaa ja nykyaikaista kuvankäsittelyä.

Kuvat toimivat havainnollistavana, viihdyttävänä ja personoivana elementtinä multimediassa. Pelkän tekstin käyttäminen tekee helposti esimerkiksi internetsivustosta tylsän ja persoonattoman. Kuvat tuovat sivustoon ilmettä ja niillä on helppo selventää tai tukea tekstissä kerrottuja asioita. (Keränen ym., 2003, 64.)

Digitaalinen kuva eli grafiikka muodostuu kuvapisteistä eli pikseleistä. Jokaisella pikselillä on oma väriarvonsa. Lopullinen arvo muodostuu kolmen arvon, punaisen, vihreän ja sinisen yhdistelmästä. Tässä järjestelmässä (RGB; red, green, blue) jokaiselle värille annetaan arvo 0 ja 255 väliltä. Jos kaikilla kolmella värillä tuo arvo on nolla, on pikseli valkoinen. Jos kaikki saavat arvon 255, on pikseli musta. (Keränen ym. 2003, 65, 67.)

Pikselikoolla tarkoitetaan esimerkiksi kuvassa tai näytöllä olevien pikselien määrää. Digitaalisella kameralla otettu kuva voi olla esimerkiksi 2048 x 1536 pikseliä, eli 2048 pikseliä leveä ja 1536 pikseliä korkea. Kuvan resoluutiolla tarkoitetaan sitä, kuinka monta pikseliä mahtuu yhdelle tuumalle. Suuri resoluutio tarkoittaa siis sitä, että pikseleitä on tiheässä ja kuva on tarkka, koska pieniä yksityiskohtia mahtuu mukaan enemmän matalampaan resoluutioon verrattuna. (Keränen ym. 2003, 66.)

Grafiikka voidaan jakaa kahteen luokkaan, bittikarttagrafiikkaan ja vektorigrafiikkaan. Bittikarttagrafiikka muodostuu pikseleistä. Esimerkiksi digitaaliset kamerat ja skanneri tuottavat bittikarttagrafiikkaa. Tällaista grafiikkaa voidaan käsitellä esimerkiksi Adobe Photoshop tai Paint Shop Pro -ohjelmistoilla. Bittikarttagrafiikan huono puoli on siinä, että kuvaa suurennettaessa siihen joudutaan lisäämään ylimääräisiä pikseleitä. Tästä seuraa se, että kuvan laatu kärsii. Samoin, jos kuvaa pienennetään, menetetään pikseleistä osa. Tämä ei sinänsä ole ongelma, mutta jos

tuota pienennettyä kuvaa halutaankin isontaa, ei menetettyä informaatiota saada enää takaisin (Keränen ym. 2003, 66).

Vektorigrafiikka poikkeaa bittikarttagrafiikasta siinä, että se muodostuu pikseleiden sijaan objekteista, jotka noudattavat matemaattisesti määriteltyjä muotoja, kuten viivoja, ympyröitä, neliöitä jne. Kun piirretään esimerkiksi ympyrä, siitä tallentuvat tiedot ympyrän sijainnista koordinaatteina, sen halkaisija ja väri, sen sijaan, että tallennettaisiin jokainen kuvassa oleva pikseli. Tämän ansiosta tiedostot ovat pieniä, ja mikä tärkeintä, vektorigrafiikka on täysin koon ja muotonsa puolesta muokattavissa, ilman että kuvanlaatu kärsii. Tämän ansiosta vektorigrafiikka on aina tarkkaa. Näytöllä näkyvä kuva on vain esityskuva siitä, mitä tallennettu matemaattinen informaatio pitää sisällään. Tunnettuja vektorigrafiikkaa käyttäviä ohjelmia ovat esimerkiksi Adobe Illustrator ja Adobe Flash. Vektorigrafiikalla tehty kuva voidaan tallentaa myös bittikarttagrafiikaksi. (Keränen ym. 2003, 70–71.)

Kuvatiedostojen koko on merkittävä tekijä esimerkiksi internetsivujen käytettävyyden kannalta. Ison kuvan lataaminen vie pidempään ja saa käyttäjän helposti turhautumaan. Tiedoston kokoon vaikuttavat kuvan pikselimäärä ja värisyvyys. Bittikarttakuvan tiedostokoko saadaan laskettua kaavasta:

$$\frac{\text{KORKEUS} \times \text{LEVEYS (pikseleinä)} \times \text{VÄRISYVYYS (bitteinä)}}{8} = \text{tiedoston koko tavuina}$$

Kuvatiedoston kokoa voidaan pienentää myös niin, että kuvan koko ja värisyvyys pysyvät samana. Tätä toimenpidettä kutsutaan pakkaamiseksi. Koska ihmissilmä ei pysty erottelemaan kuin tietyn määrän värejä, voidaan osa väritiedosta hävittää. Pakkaaminen on tehokkainta sellaisten kuvien tapauksessa, joissa on paljon yksityiskohtia ja sävyjä, esimerkiksi valokuvat ovat tällaisia. Sen sijaan kuvankäsittelyohjelmalla piirrettyssä Suomen lipussa on vähemmän pakattavaa. Kuvaa pakattaessa sen laatu putoaa ja jos kuvaa pakataan paljon, se ei enää näytä samalta, vaan muuttuu epäteräväksi ja laatikkomaiseksi. (Keränen ym. 2003, 68.)

Kuvan käyttötarkoitus ratkaisee sen, mihin tiedostomuotoon se halutaan tallentaa. Tiedostomuotoja on useita ja niillä on omat hyvät ja huonot puolensa. Yleensä ohjelmat osaavat sekä lukea, että tallentaa useita eri tiedostomuotoja. Multimedia-tuotannossa käytetään yleensä TIFF-, BMP- ja JPEG-formaatteja. Internetsivuilla voidaan käyttää ainakin GIF-, PNG- ja JPEG-formaatteja. Tarkempi vertailu eri formaattien välillä löytyy taulukosta 1. (Keränen ym. 2003, 72-74; CIVISION Technologies; wikipedia hakusanoina tiedostomuodot).

**Taulukko 1.** Kuvatiedostomuodot

Tiedostomuoto	Häviöllisyys	Käyttötarkoitus / ominaisuudet
JPEG	häviöllinen	Yleisin, sopii erityisesti valokuviin. Aina täydet, 24 bitin väritiedot. Ei voida tallentaa läpinäkyvää väriä.
GIF	häviötön	Vain 256 väriä, ei ole hyvä valokuville. Sopii parhaiten kuville, joissa vähän sävyjä ja yksityiskohtia, kuten piirrutyksille ja kaavioille. Voidaan tehdä helposti animaatioita.
TIFF	valittavissa	Suosittu ammattikäytössä. Etuna kyky tallentaa useita kuvia samaan tiedostoon. Voidaan käyttää häviöttömänä, eli muokata ja tallentaa uudelleen laadun kärsimättä. Joitain yhteensopivuus ja tietoturvaongelmia.
PNG	häviötön	GIF:n ilmainen vaihtoehto. Tosin tukee myös täysvärikuvia. Vanhat selaimet eivät tue
BMP	häviötön	MS Paintin tiedostomuoto, käyttää 1 - 24-bittistä värisyvyyttä. Voidaan pakata häviöttömästi.
PDF	häviötön tekstin osalta, häviöllinen kuvien, videon ja äänen osalta	Perustuu PostScript-sivunkuvauskielen. Soveltuu erityisesti sähköiseen julkaisemiseen, tulostamiseen ja painamiseen. Tiedosto aukeaa kaikilla koneilla ja käyttöjärjestelmillä. Tiedostoon voi laittaa linkkejä tai äänitiedostoja jotka alkavat sivun avautuessa. PDF-tiedoston sisältämästä tekstistä voi tehdä hakuja.

### 2.2.3 Ääni

Ennen digitaalista äänen tallennusta ja muokkausta oli äänen käsittely huomattavasti hankalampaa ja harvempien ulottuvissa olevaa toimintaa. Nykyisin jokainen voi helposti tallentaa ja muokata lähes mitä tahansa ääntä omalla tietokoneellaan.

Multimediassa ääntä voidaan käyttää musiikin, puheen ja tehosteiden muodossa. Musiikin valinnassa tulee olla tarkkana, koska esimerkiksi sama video saa eri merkityksen pelkästään taustamusiikkia vaihtamalla. Musiikilla voidaan saada aikaan monta eri asiaa tai vaikutusta. Musiikki voi toimia viihdyttävänä element-

tinä tai tunnelman luojana. Toisaalta musiikki voi toimia myös informaation lähteenä. Jos esimerkiksi soitetaan suomalaista 1940-luvun musiikkia ja näytetään mustavalkoista kuvaa, niin aika ja paikka on kuulijalla selvä jo ennen kuin paikkaa tai aikaa on mainittu tai kuvilla näytetty. Samalla myös tietty tunnelma on luotu kuulijalle. Toinen esimerkki voisi olla vaikkapa intialaisen musiikin soittaminen. Mielikuva maasta, kielestä ja kulttuurista välittyy heti ja tehokkaammin, kuin vaikkapa puheesta tai tekstistä. Toisaalta musiikkia voidaan käyttää, kuten logoa. Moni muistaa Burana-mainoksen kappaleen ja yhdistävät sen helposti kyseiseen tuotteeseen, vaikka kyseinen kappale ei edes ole alun perin kyseistä tuotetta varten tehty. Musiikin käyttöä tulee aina miettiä tarkkaan. Käyttäjä ei välttämättä halua häiritseviä ääniä, tai voi olla, että hänellä on kaiuttimien tai kuulokkeiden äänet kovalla tai poissa kokonaan. Toisaalta hänellä voi olla esimerkiksi omaa musiikkia tai muuta ääntä taustalla, jolloin hän joutuu pysäyttämään jommankumman. Esimerkiksi internetsivuilla musiikki yleensä ärsyttää käyttäjää enemmän kuin miellyttää. (Keränen ym. 2003, 80; Keränen ym., 2006, 192.)

Miettimisen arvoisia asioita musiikin käyttämisen suhteen ovat edellä mainittujen lisäksi se, käytetäänkö instrumentaalista musiikki vai laulettua. Myös sanojen merkitystä kannattaa miettiä ja niiden sopivuutta muuhun sisältöön. Musiikin tyylin lajin ja temmon tulee olla harmoniassa muun materiaalin kanssa. (Keränen ym. 2003, 80–82.)

Puheella voidaan lisätä multimedian muuhun sisältöön tai sillä voidaan tukea muuta sisältöä. Puhuja voi kertoa lisätietoa tai johdattaa kuuntelijaa ”esityksen” läpi. On muistettava, että puhutun kielen ja tyylin tulee sopia kohdeyleisölle ja aiheeseen, aivan kuten musiikinkin. Kuulija ei kuule pelkästään sanotut sanat, vaan myös puhujan sukupuolen, murteen, äänenpainot, puhetyylin jne. Puheen tulee olla selkeätä, eikä siinä kannata esittää liian monimutkaisia asioita tai suuria lukuja. Puhuttujen osuuksien tulisi pysyä alle minuutin mittaisina. Jos ääntä on pitkästi, tulisi se voida keskeyttää. Aivan kuten video voidaan ”pause”-toiminnolla. (Keränen ym. 2003, 82.)

Tehosteet tukevat kuvaa. Niillä voidaan lisätä tunnelmaa tai tukea kuvan, videon tai tekstin luomaa tunnelmaa. Esimerkiksi jos näytetään videota kesäisestä järvestä, voidaan taustalle lisätä linnunlaulua tai tuulenvirettä, vaikka niitä ei alkuperäisessä videotiedostossa olisikaan. Tehosteilla voidaan korostaa toimintaa, esimerkiksi jos ruudussa on liikettä, sitä voidaan korostaa sopivalla äänitehosteella. Tehosteet sopivat hyvin myös käyttäjän toiminnan todentamiseen. Esimerkiksi hiiren painalluksesta voi seurata jokin ääni. (Keränen ym. 2003, 81–82.)

Äänitehosteet voidaan jakaa kahteen ryhmään, synkronistehosteisiin ja pistetehosteisiin. Synkronistehoste kuuluu liikkeen aikana ja on osa sitä. Esimerkiksi auto pitää ääntä liikkessaan tai ihminen kävellessään kengät jalassa. Pistetehoste ovat ääniä, joiden tarkoitus on kiinnittää käyttäjän huomio esimerkiksi kuvaan. Esimerkkinä voisi olla vaikkapa seikkailupeli, jossa hellalla oleva teepannu rupeaa viheltämään. Pelaaja kiinnittää automaattisesti huomion tuohon ääneen ja voi arvata, että sillä on jokin merkitys tai käyttötarkoitus pelissä. (Keränen ym. 2003, 81–82.)

Erityyppisistä äänistä voitaisiin mainita vielä reaktioäänet ja puhesyntetisaattorit. Reaktioäänillä tarkoitetaan ääniä, jotka antavat käyttäjälle palautetta sovelluksen toiminnasta. Esimerkiksi painiketta painettaessa voi tulla ääni, joka kertoo käyttäjälle, että nyt on painettu painiketta. Varoitusaänet ovat myös reaktioääniä. Puhesyntetisaattoreita käytetään harvoin multimediassa. Autoissa käytettävissä navigaattoreissa tätä monesti käytetään, koska käyttäjä ei voi katsoa näyttöä ajaessaan. (Keränen ym. 2003, 81–82.)

Äänestä ja äänitiedostoista on hyvä tietää perusteet, kun multimediaa suunnitellaan ja tehdään. Kaiuttimia on yleensä kaksi, joten mono- ja stereoäänten erot tulee ymmärtää. Monoääni ei anna tilavaikutelmaa, koska sama ääni tulee molemmista kaiuttimista. Stereoäänessä kuullaan eri kohdista äänitetty ääni, jolloin molemmista kaiuttimista tulee hieman eri ääni. Yleensä puhe äänitetään monona ja musiikki stereona. (Keränen ym. 2003, 82.)

Äänen resoluutiolla tarkoitetaan käytännössä sitä, kuinka monella eri voimakkuusarvolla ääni voidaan esittää. Yleisimmät resoluutiot ovat 8 ja 16 bittiä. 16-

bittinen kuulostaa virheettömältä verrattuna heikompaa 8 bitin resoluutioon. Äänen taajuus ilmaistaan hertseinä (Hz). Hertsi ilmaisee sen, että kuinka monta kertaa ääni värähtelee yhden sekunnin aikana. Mitä matalampi ääni on, sitä matalampi on arvo hertseinä. Korkea ääni taas saa korkeamman hertsiarvon. (Keränen ym. 2003, 83.)

Äänen pakkaus tapahtuu koodekin avulla. Koodekilla tarkoitetaan algoritmiparia, joista toinen pakkaa ja toinen purkaa äänitiedoston alkuperäiseen muotoonsa. Koodekkeja on useita ja ne liittyvät käytettävään tiedostomuotoon. Koodekkeja ovat yleensä ohjelma- ja valmistajakohtaisia. Taulukossa 2 on yleisimpiä tiedostomuotoja ja niiden pakkausmenetelmät. (Keränen ym. 2003, 86.)

#### **2.2.4 Video**

Digitaalista videota voidaan käyttää digitaalisessa tv-lähetyksessä, internetissä, CD-, DVD- ja Blue-Ray-levyillä, peleissä sekä erilaisissa multimediaesityksissä. Videon tallentaminen digitaaliseen muotoon on tietotekniikan kehityksen kanssa mahdollistanut videon luomisen, muokkaamisen ja käyttämisen helpottumista ja nopeutumista. Sopivat ohjelmat hankkimalla kuka tahansa voi tehdä laadukasta videoeditointia, johon olisi vielä pari vuosikymmentä sitten tarvittu kalliit laitteet.

Katsotaan videota missä tilanteessa tai paikassa tahansa ja miltä laitteelta tahansa, sen toteutuksessa tulee käyttää elokuvatuotannosta tuttuja tapoja, tekniikoita ja kerrontaa. Perinteinen elokuva tai videoesitys on passiivista viestintää. Katsoja ei osallistu itse mitenkään sisältöön tai sen kulkuun. Tietokonepeleissä ja muissa tietokoneella katsottavissa multimediatuotoksissa on kuitenkin mahdollista antaa käyttäjälle kyky vaikuttaa videon kulkuun ja sisältöön. (Keränen ym. 2003, 92.)

Video on tehokas tapa viestiä. Sillä on helppo saada käyttäjän huomio ja mielenkiinto. Pelkän tekstin sijaan esimerkiksi videon näyttäminen tarjoaa monipuolisempaa ärsykettä liikkuvan kuvan ja äänen ansiosta. Opiskelija näkee mitä pitää tehdä, sen sijaan, että hänen tulisi ymmärtää toisen ihmisen tekstimuotoinen kuvaus aiheesta. Videossa liikkuva kuva saadaan yhdistettyä ääneen, esimerkiksi puhujan ääneen, jolloin tiedon kulku kuulijalle tehostuu entisestään esimerkiksi

pelkkään tekstiin nähden. Ihmisen pääasialliset aistit tiedon keruuseen ovat näkö- ja kuuloaisti. Molempia kannattaa hyödyntää. Oma kokemukseni on, että etäopiskelu videoiden avulla on selvästi tehokkaampaa ja miellyttävämpää kuin tekstin lukeminen. (Keränen Lamberg & Penttinen 2006, 205.)

Tapahtumien kulun esittäminen videolla tai haastattelun näyttäminen vakuuttavat käyttäjän helpommin, kuin pelkkä luettu tieto. Videolla voidaan myös helpohkosti välittää haluttu tunnelma ja mielikuva. Esimerkiksi esityksen introna näytetty video sopivalla musiikilla höystettynä saadaan katsojan huomio ja tehokkaampi vaikutus kuin pelkän puheen, tekstin tai kuvien avulla. (Keränen ym. 2003, 92; Keränen ym. 2006, 205.)

Videon käyttämistä mietittäessä tulee aina huomioida kohderyhmä ja kohderyhmän käyttämä media. Kohderyhmä vaikuttaa videon sisältöön, rakenteeseen ja tyyliin. Videon laatuun ja kokoon vaikuttaa käytettävä media. Matkapuhelimeen tarkoitettu video on ruutukooltaan ja tiedostokooltaan pienempi kuin esimerkiksi DVD-levylle laitettava video. Tulee siis aina huomioida käyttäjän käyttämän laitteen prosessointikyky, näytön koko, sekä tiedonsiirron nopeus verkosta tai käytettävissä ole tallennuskapasiteetti. Mobiililaitteista puhuttaessa täytyy myös muistaa tiedonsiirron korkea hinta. Jos ei ole tarvetta näyttää tapahtumien kulkua tai liikettä, voi pelkkä kuva olla parempi vaihtoehto. Kuvaan saadaan helposti iso resoluutio ja tiedostokoko pysyy pienempänä kuin videossa. Kuvasta on helpompi hahmottaa esimerkiksi kaavio tai muita pieniä yksityiskohtia sisältäviä näkymiä kuin videosta. Vaikka video onkin tehokas viestinnän keino, tulee sen käytölle löytyä perustelut.

Verkossa videota voidaan jakaa joko suoratoistona (streaming) tai oman koneen kiintolevylle tallennettavana tiedostona. Suoratoistossa on se etu, että videon katselu voidaan aloittaa heti. Kiintolevylle ladattava tiedosto pitää ladata kokonaan, ennen kuin sitä voidaan ruveta katselemaan. (Keränen ym. 2003, 94–95.)

Digitaalista videota tehtäessä tulee miettiä ruudun koko, ruutujen määrä sekunnissa (FPS, frames per second) ja käytettävä pakkaus. Kuten digitaalisessa kuvassa, myös digitaalisessa videossa kuvan koko ilmoitetaan resoluutiona, eli montako

pikseliä on pysty ja vaakatasossa. Esimerkiksi kuva voi olla 640 x 480 pikseliä, joka on suhdelukuna ilmaistuna 4:3. Tätä kutsutaan kuvasuhteeksi. Tavallinen televisio on 4:3:n ja laajakuvatelevisio 16:9:n.

Videokuva koostuu yksittäisistä kuvista, joita näytetään tietty määrä sekunnissa. Euroopassa televisiot näyttävät 25 ruutua sekunnissa. Mitä suurempi määrä ruutuja sekunnissa, sitä isompi on tiedosto koko ja sitä sulavampaa on videolla näkyvä liike. Ihmissilmä ei kuitenkaan huomaa merkittävää eroa tietokoneruudullakaan, vaikka kuvia esitettäisiinkin yli 30 sekunnissa. (Keränen ym. 2003, 100.)

Videotiedostot pakataan soivaan formaattiin tiettyä koodekkia, eli pakkausohjelmistoa käyttäen. Tämän jälkeen niitä voidaan katsella videon toistoon tarkoitetulla ohjelmalla. Pakkauksesta syntyvän lopullisen videotiedoston koko määräytyy käytettävästä ruudun koosta, ruutujen määrästä sekunnissa, videon pituudesta sekä pakkauksen määrästä. Videon toistoon vaadittavaan tiedonsiirtoon vaikuttavat edellä mainitut videon pituutta lukuun ottamatta. Tarvittava tiedonsiirtonopeus pakkaamattomalle videolle voidaan laskea seuraavasti.  $640 \times 480 \times 25 \times 24 = 69\,120\,000$  bittiä sekunnissa, eli n. 23 megatavua sekunnissa. 640 x 480 tarkoittaa resoluutiota, 25, ruutuja sekunnissa ja 24 viittaa värisyvyyteen. Pakkaaminen pienentää tiedostokokoa ja siis myös tarvittavaa siirtonopeutta hyvinkin paljon, laadun merkittävästi vielä kärsimättä (Keränen ym., 2003, 100–102). Käytetyimmät videotiedostomuodot on listattu taulukossa numero 2. (Keränen ym. 2003, 99, 100; FileInfo.com, 2010.)



**Taulukko 2.** Videotiedostomuodot

Tiedosto muoto	Tiedos- topääte	Käyttötarkoitus / ominaisuudet
AVI	.avi	Windows ympäristöön suunniteltu videotiedostotyyppi. Hyviä editointikäyttöön. Katseluohjelman tulee tukea videon pakkaamiseen käytettyä koodekkia.
Quick-Time	.mov / .Moov	AVI:n vastine Macintoshille. Voidaan katsoa myös PC:llä QuickTime:n PC-versiolla.
MPEG	.mpg	MPEG on videolla ja äänelle suunniteltu standardi, joka pitää sisällään useita tiedostomuotoja, eri tarkoituksiin.
RealMedia	.rm	Suoratoistoon, eli tietoverkon välityksellä jaettavan äänen ja videon jakamiseen tarkoitettu formaatti. Tiedostoa ei voida editoida tai pakata sen luomisen jälkeen.
MP4	.mp4	MPEG-standardin yksi jäsen. Pakkaa videon MPEG-4 pakkauksella ja äänen AAC-pakkauksella. Suosittu tiedostomuoto esimerkiksi internetissä (YouTube).
WMV	.wmv	Microsoftin kehittämä tiedostotyyppi erityisesti tietoverkko jakelua varten.
FLV	.flv	Flash yhteensopiva videotiedostotyyppi. Voidaan luoda Flashilla tai muilla ohjelmilla, jotka voivat luoda Quick-Time-tiedostoja.

## 2.3 Väylät

Multimedia tarjotaan käyttäjälle aina jotain tiettyä ”väylää” pitkin. Väylällä tarkoitetaan sitä laitetta tai tallennetta, jossa multimedia tarjotaan käyttäjälle. Tällaisia voivat olla esimerkiksi CD, DVD, Blu-ray, internet, lähiverkko, televerkko, sekä radio- ja televisioverkot. Jokainen näistä vaatii tiettyjen asioiden huomioimista. Jo suunnitteluvaiheessa tulee huomioida, mikä on käytettävä väylä, koska se vaikuttaa itse multimediatuotokseen. Jos jakelukanavana käytetään esimerkiksi matkapuhelinverkkoa, tulee huomioida, että laite jota suurin osa käyttäjistä käyttää, on matkapuhelin tai muu kannettava laite, jolla on hidas kirjoittaja ja jossa on pieni näyttö. Myös käytettävän väylän tiedonsiirtonopeus on olennainen tekijä multimediaa suunniteltaessa. (Keränen Lamberg & Penttinen 2006, 118–119.)

CD eli Compact Disc on erittäin laajasti käytetty tallennemuoto. Sitä voidaan käyttää lähes kaikilla tietokoneilla, joskin joissain uusimmissa Macintosh-koneissa ei ole enää CD-asemaa. CD-levylle voidaan tallentaa noin 650 megatavua tietoa. CD:n kapasiteetti ja tiedonsiirtonopeus ovat yleensä riittäviä melko vaativallekin multimedia sisällölle, ellei tarkoitus ole käyttää pitkiä tai todella korkealaatuisia videoita. CD:n huono puoli verrattuna DVD:hen on sen pienempi kapasiteetti, sekä se, että erillisillä DVD-soittimilla ei pysty katselemaan CD:tä. Myös CD:n tiedonsiirtonopeus on alempi kuin DVD:llä. (Keränen Lamberg & Penttinen 2006, 118–119.)

DVD eli Digital Versatile Disc on myös erittäin yleinen tallennemuoto. DVD-asema löytyy kaikista uudehkoista tietokoneista, myös kaikista Macintosheista. Myös erilliset DVD-soittimet ovat suosittuja. Tosin jos DVD tehdään tällaiselle laitteelle pelkän tietokonekäytön sijaan, tulee se ottaa huomioon jo suunnitteluvaiheessa. Suurin ero CD:hen verrattuna on DVD:n selvästi suurempi kapasiteetti. DVD:tä valmistetaan yksi ja kaksipuolisina sekä yksi- tai kaksikerroksisina. Kapasiteetti on käytettävistä mallista riippuen 4,7 gigatavusta aina 17 gigatavuun asti. Tämä mahdollistaa suurten ja laadukkaiden videotiedostojen käytön. DVD:n yleisin käyttötarkoitus onkin viihdealalla, elokuvien myynnissä ja vuokrauksessa (Keränen ym. 2003, 118–119)

Kaikkien edellä mainittujen, fyysisesti jaettavien levyjen ongelmana on sisällön kankeus; niitä ei voida päivittää, eikä todellista vuorovaikutusta valmistajan tai muiden käyttäjien ja käyttäjän välillä pääse syntymään. Tässä suhteessa tietoverkot, eteenkin internet, on ylivoimainen. Verkkojen ongelmana on tiedonsiirtonopeus, joka aiheuttaa monesti käyttäjälle odottamista, jos jaettavat tiedostot vaativat suurta tiedonsiirtonopeutta reaaliajassa toimiakseen. (Keränen Lamberg & Penttinen 2006, 118–119.)

Verkkomultimedialla tarkoitetaan tietoverkossa jaettavaa multimediaa, kuten ääntä, videota, kuvia tai animaatioita. Tyypillistä verkkomultimediaa ovat www-sivustot ääni, kuva ja video tiedostoineen sekä multimediaohjelmat, pelit ja hyötyohjelmat. Myös radio- ja tv-lähetykset internetissä ovat tyypillistä verkkomulti-

mediaa. Mainitsemisen arvoisia ovat myös erilaiset virtuaaliympäristöt ja todellisuudet, kuten Google maps -karttapalvelu tai Habbo Hotelli, joka on virtuaalinen ympäristö, jossa käyttäjä esiintyy muille käyttäjille luomansa virtuaalisen hahmon kautta.

Verkossa olevaa multimediaa katsellaan ja kuunnellaan www-selaimeen asennettun laajennuksen avulla tai tietokoneelle asennetulla erillisellä katseluohjelmalla, kuten RealPlayer tai Windows Media Player. Jos halutaan enemmän vuorovaikutusta, kuten esimerkiksi pelien ja multimediasovellusten tapauksessa, voidaan käyttää esimerkiksi Flashillä toteutettua vuorovaikutteista sovellusta, joka toimii selaimessa.

Ääntä ja videota voidaan jakaa ladattavina tiedostoina tai suoratoisto-tekniikkaa käyttäen. Esimerkiksi video voidaan ladata tietokoneen kovalevylle ja katsella sieltä. Suoratoistolla taas videon katselu voidaan aloittaa heti, videon samalla latautuen. Tämä tietysti vaatii sitä, että video latautuu vauhdilla, joka vie vähemmän aikaa kuin videon katselu. Muuten katsoja joutuu odottelemaan videon latautumista. Suoratoisto on käyttäjäystävällisempää, koska katselu voidaan aloittaa heti sen sijaan, että odotetaan mahdollisesti isonkin tiedoston latautumista loppuun asti. Käyttäjän verkkoyhteyden nopeus määrittää sen, kuinka laadukasta videota hän voi suoratoistaa ilman, että video joudutaan välillä keskeyttämään (Keränen ym. 2006, 103–104).

Televerkkojen osalta keskitymme matkapuhelinverkkoihin. Matkapuhelinverkko on langaton, lähes joka puolella toimiva verkko. Niistä yleisin multimedia sovellusten käyttämiseen soveltuva on UMTS (Universal Mobile Telecommunications System), joka on kolmannen sukupolven (3G) matkapuhelinverkko. Tällaista verkkoa käytetään enimmäkseen matkapuhelimilla ja kämmen tietokoneilla. Alueella, jossa kenttä on vahva, voidaan käyttää esimerkiksi internetiä aivan normaalisti sivustojen selaamiseen, videoiden katseluun ja pelien pelaamiseen. 3G-verkot eivät kuitenkaan kata esimerkiksi koko Suomea ja signaalin vahvuudessa on eroja. Tämä vaikuttaa siirtonopeuteen. Monista internetsivuista on mobiili-versiot, jotka ovat yksinkertaisia versioita normaalista internetsivuista. Niissä on esimerkiksi

lähinnä teksti sisältö, kuvat ja videot on voitu poistaa käyttämisen helpottamiseksi ja nopeuttamiseksi.

Suunniteltaessa mobiilisovellusta, eli multimediaa, jota jaetaan matkapuhelinverkoissa, tulee huomioida erityisesti siirtonopeuden rajallisuus, päätelaitteen pieni näyttö sekä näiden laitteiden käyttöjärjestelmien standardit ja mitä tiedostotyyppiä ne pystyvät toistamaan (Keränen, ym., 2003, 124-126) . WAP (Wireless Application Protocol) on mobiilipalveluja varten kehitetty joukko protokollia, jonka ohjelmointimalli pohjautuu WWW-ohjelmoinnin vastaavaan. MMS (Multimedia Messagin Services) on viestipalvelu standardi, joka mahdollistaa multimediamessagien lähettämisen eri valmistajien matkapuhelimien välillä. MMS on tehty matkapuhelinten välillä käytettäväksi, mutta multimediamessagi voidaan lähettää myös puhelimesta sähköpostiin ja päinvastoin. MMS tukemat tiedostomuodot ovat tietokonemaailmasta tuttuja. Jotta näitä palveluja voidaan käyttää, tulee matkapuhelimen tukea kyseistä palvelua. Vanhat puhelinmallit eivät tue 3G-verkon tiedonsiirtoa tai MMS-viestejä (Ketamo, Multisilta 2004, 229–234).

## 2.4 Ohjelmistot ja formaatit

Multimedian luomiseen tarvitaan erilaisia ohjelmistoja. Sopivien ohjelmien valinta riippuu tuotettavan multimedian tyypistä. Sisältääkö se tekstiä, ääntä, kuvia, videota, animaatioita tai jotain näiden yhdistelmää. Myös kanava, jonka kautta tuotetta aiotaan jakaa vaikuttaa. Tuleeko se CD:lle, vai internetsivuille. Ohjelmistojen valintaan vaikuttaa myös tekijän käytettävissä oleva kalusto. Esimerkiksi onko käytössä Applen tietokone vai PC. Jos halutaan tehdä todella korkealaatuista ja ammattimaista jälkeä, tulee ohjelmiston olla sitä tasoa. Jos taas ns. harrastelijatason ohjelmisto tarjoaa riittävät työkalut, voidaan sellaistaakin käyttää. Jottei valinnan tekeminen olisi liian helppoa, voidaan lisätä vielä yksi muuttuja, nimittäin se, että onko ohjelmisto kaupallinen vai open source, eli vapaaseen lähdekoodiin perustuva.

Multimedian tekoon voidaan hakea ohjelmistoja monelta eri valmistajalta tai open source-pohjalta, tai sitten voidaan valita ison valmistajan tuotepaketti, joka pitää sisällään kaikki tarvittavat ohjelmat. Multimediaohjelmistojen markkinajohtajalta

Adobelta löytyy ohjelmistot multimedian joka osa-alueelle. Tämä saattaa olla yritykselle helpoin ja toimivin ratkaisu, joskaan ei välttämättä halvin.

Flash on vektorigrafiikkaan perustuva multimediamyöristö. Se on kehitetty ensisijaisesti multimedian ja animaatioiden verkkojakeluun. Flashia käytetään nykyisin laajalti myös internetsivujen tai niiden osien tekemiseen, sekä esimerkiksi internetpelien ja muiden verkkosovellusten toteutuksessa. Flashissa kaikki grafiikka paitsi valokuvat ovat vektorigrafiikkaa. Vektorigrafiikan etuna on se, että sitä voidaan skaalata ja liikuttaa vapaasti ilman laadun heikkenemistä. Flash-ohjelmaan sisältyy ActionScript-ohjelmointikieli, joka mahdollistaa erittäin monipuolisten ja joustavien sovellusten tekemisen. Flash-tiedostojen katselemiseen tarvitaan Flash Player -sovellus, joko selainlaajennuksena tai itsenäisenä sovelluksena. Flash on alun perin Macromedian kehittämä, mutta Adoben Systemsin ostettua Macromedia vuonna 2005, siirtyi omistus Adobelle. (Keränen ym. 2003, 107; Ketamo ym. 2004, 125–126.)

Flash sovelluksia luodaan ohjelmalla, jonka uusin versio on Flash CS5. Flash toimii aikajanapohjaisesti. Aivan, kuten video koostuu useista kuvista, koostuu flash elokuva yhdestä tai useammasta kehyksestä (frame). Toistonopeus, eli se montako kehystä edetään sekunnissa, voidaan määritellä halutun suuruiseksi. Kaikki näkyvä sisältö sijoitetaan haluttuihin kehyksiin. Sisältö voidaan jakaa usealle tasolle (layer). Jokainen taso on oma aikajanansa, eikä katsoja näe eroa sen suhteen, mikä objekti on milläkin aikajanalla. Usean aikajanien käyttäminen helpottaa kuitenkin selvästi monimutkaisten animaatioiden toteutusta. ActionScriptin avulla voidaan aikajanalla hyppiä tai pysähtyä ja jatkaa taas aivan miten tekijä haluaa. Tekijä näkee näyttämöllä (stage) kuvan siitä, mitä sillä hetkellä valittuna olevassa kehyksessä on. Kaikki objektit sijoitetaan näyttämölle, kunhan oikea kehys on ensin valittuna. Kehyksiä on kahta perustyyppiä, tavallisia ja key, eli avainkehyksiä. Avainkehyksiin sijoitetaan varsinaiset alku- ja lopputulokset, sekä muutokset. Niiden välissä olevat tavalliset kehykset pitävät sisällään saman sisällön, joka on edellisessä key-kehyksessä. Poikkeuksena tähän ovat erilaiset liikkeen, reitin ja muodonmuutoksen animointiin liittyvät tekniikat, joista lisää seuraavaksi.

Liikeanimaation tekeminen on varsin helppoa Flashilla. Jokaisen kehyksen sisältö voidaan luoda itse käsin, joka on tosin työlästä ja hidasta. Helpompi vaihtoehto liikeanimaation tekemiseen on motion tweening. Siinä avain kehyksiin, eli key-kehyksiin sijoitetaan halutun animaation alku- ja lopputulos. Flash animoi tarvittavan muutoksen välissä oleviin kehyksiin automaattisesti. Muotojen muutokseen voidaan käyttää shape tweeningä, joka on vastaava toiminto mutta objektien liikkeen sijaan se on tarkoitettu objektien muodonmuutoksia varten. Kolmas animointia helpottava toiminto on motion path -animointi, jolla pystytään antamaan objektille haluttu liikerata, jota se sitten seuraa. Kaiken tämän lisäksi voidaan animaatiota ohjelmoida ActionScriptin avulla. (Ketamo ym. 2004, 127-129.)

Flash-tuotokseen voi liittää monia eri video-, ääni- ja kuvatiedostoja. Kaikki ääni pakataan oletusarvoisesti MP3- ja bittikartat JPEG -tiedostoiksi. Lopullinen selaimen liitettävä tiedosto on .swf. Sitä voidaan toistaa selaimen lisäksi erillisellä selaimen ulkopuolisella Flash Player -sovelluksella. Flashilla voidaan tehdä myös esimerkiksi itsenäisesti ajettava .exe -tiedosto, Macintosh Projector -tiedosto sekä .mov ja .html -tiedosto. (Ketamo ym. 2004, 128.)

Flashin käytön ongelmia www-sivusto -käytössä voivat olla niiden raskaus ja huono soveltuvuus mobiililaitteiden selaimilla käytettäväksi. Ongelmia voi muodostaa myös se, etteivät hakukoneet osaa välttämättä indeksoida Flash-esityksissä olevaa tekstisisältöä, etenkään jos se on haettu ulkoisesta tiedostosta. Kokonaan Flashilla toteutettujen sivustojen ongelmana on se, ettei muusta kuin pääsivusta voida antaa linkkiä, koska kaikki sisältö on itse Flash -tiedostossa. (Ketamo ym. 2004, 127-129.)

ActionScript 3 on ammattilaiskäyttöön suunniteltu tehokas ja monipuolinen ohjelmointikieli. Sillä voidaan ohjata käytännössä kaikkia animaation objekteja, ääntä, grafiikkaa ja videota. AS3:n toiminta ei rajoitu vain Flash-animaation ohjailuun, vaan sillä voidaan myös hakea, viedä, jäsennellä ja muokata tietoa ulkopuolisista tiedostoista. Flashilla voidaan hakea ja viedä tietoa XML-tiedostoista, web-

palvelimilta ja se pystyy käyttämään suoraan TCP-yhteyksiä. AS3 on osittain oliopohjainen ja hyvin samankaltainen JavaScriptin kanssa.

## 2.5 Multimediatuotteen käytettävyys

Käytettävyyden voidaan sanoa koostuvan opittavuudesta, muistettavuudesta, tehokkuudesta, virheettömyydestä ja miellyttävyydestä. Jos multimediatuote on tehty hyvin, ei sen toimintaa tarvitse välttämättä varsinaisesti edes opetella tai muistaa, ainakaan ulkoa. Tuotteen pitäisi olla niin selkeä ja siinä tulisi käyttää niin yleismaailmallista symboliikkaa ja niin selkeää kieltä ja ohjeistusta, että käyttäjä pystyy pienellä pohdiskelun jälkeen päättämään, mitä sovelluksessa voi tehdä, mistä mikäkin löytyy ja mitä mistäkin toiminnosta seuraa. Tietotekniikkaa käyttäneille on muodostunut käsitys yleisistä symboleista ja esimerkiksi siitä, miten linkit tai painikkeet eroavat muusta sisällöstä. Tämän tyyppistä perustietoa tulee hyödyntää. (Ketamo ym. 2004, 81–83.)

Tehokkuutta voidaan parhaiten mitata sillä, kuinka tehokkaasti kokenut käyttäjä pystyy hyödyntämään järjestelmää. Tämä on tavallaan tuotteen esteettisyyden vastakohta. Tehokas ei välttämättä ole kaunista. Liika panostaminen esteettisyyteen saattaa alentaa tehokkuutta merkittävästi. Tehokkuuden kannalta olennaista on myös virheettömyys. Hyvällä suunnitellulla voidaan vähentää käyttäjän tekemiä virheitä. Mitä enemmän käyttäjä tekee virheitä, sitä huonommin suunniteltu on tuote. Virhetilanteessa käyttäjän muodostama käsitelmä ei vastaa todellista. Virheen käsittelyssä tulisi pyrkiä ohjaamaan käyttäjän käsitelmää todellisuutta vastaavaan suuntaan. Virheilmoituksen tulisi neuvoa, miten tulee toimia, jottei virhettä tule ja haluttu tehtävä saadaan suoritettua. Myös virheen syy tulisi selvittää, jotta käyttäjä oppisi ymmärtämään järjestelmää paremmin. (Ketamo ym. 2004, 83–84.)

Hyvä käytettävyys lähtee siitä, että käyttäjä otetaan huomioon. Käytettävyys tulisi huomioida jo suunnitteluvaiheessa. Käyttäjän ja hänen tarpeidensa tulisi olla sovelluskehityksen keskeisin lähtökohta, ei tekijöiden näkemys siitä, millainen tuotteen tulisi olla. Tuotteen keskivertokäyttäjä tulisi siis selvittää ja häneen tulisi tutustua tarkasti. Hänen toimintaansa tulisi tuntea ja ymmärtää.

Hyvä käsitelmä on olennainen osa käytettävyyttä. Se on tavallaan käytettävyyden kivijalka. Käyttäjällä itsellään on jokin käsitelmä päässään. Sovelluksen tulisi vastata tätä käsitelmää tai olla niin looginen, että näkemällä sovelluksen, hän voi päätellä miten asiat toimivat ja etenevät. Jos näin ei ole, on käyttäjä täysin arvailujensa varassa ja joutuu kokeilemaan miten sovellus toimii ja mitä mistäkin seuraa. (Ketamo ym. 2004, 85–87)

Käyttöliittymä ei ole vain pintakiiltoa, vain käyttäjän kosketuspinta sovellukseen. Se on siis erittäin tärkeä osa sovellusta, eikä vain jotain, joka ”täytyy tehdä”. Näkyvyys on yksi keskeinen termi käyttöliittymää suunniteltaessa. Sillä tarkoitetaan sitä, miten kaikki ominaisuudet ja toiminnot tuodaan käyttäjän tietoon. Mistä valikosta löytyy mitään, mitä kaikkea kukin painike tai säädin säätää jne. Käyttöliittymän tulee tarjota käyttäjälle tehtävän suorittamisen kannalta olennaiset tiedot. Niiden pitää olla selkeitä ja löytyä / erottua selvästi muusta tiedosta. Tätä helpottaa esimerkiksi se, että sovelluksen jokaisessa näkymässä on olennaiset toiminnot ja tiedot löytyvät samasta paikasta ja samalla tavalla sijoiteltuna. Käyttäjän pitää myös pystyä havaitsemaan oman toimintansa vaikutukset sovelluksen tilaan. Esimerkiksi jos jonkin asian lataaminen kestää muutamaa sekuntia pidempään, tulee latauksen edistymistä havainnollistaa tekstin tai grafiikan avulla. Myös painikkeen painamisen jälkeen tulisi tarjota tieto siitä, että jotain on tapahtunut, vaikka erillisellä ikkunalla, joka kertoo asiasta. Luonnollisesti jos vaikutuksen näkee muulla tavalla, ei tämä ole tarpeellista. (Ketamo ym. 2004, 90.)

Kaikki informaatio mitä ihminen näkee tai kuulee ei rekisteröidy. Hänen aistinsa keräävät enemmän informaatiota, kuin hän pystyy havaitsemaan. Kun ihminen altistetaan informaatiolle, hän valikoi sitä kolmen perusmekanismin keinoin. Nämä perusmekanismit ovat valikoiva tarkkaavaisuus, automaattinen tarkkaavaisuus ja suuntautumisrefleksi. (Ketamo ym. 2004, 90–91.)

Valikoivassa tarkkaavaisuudessa kyse on huomion tietoisesta kiinnittämisestä johonkin toimintaan tai kohteeseen. Jos ihminen haluaa tehokkaasti oppia jotain, tulee kyseessä olla valikoiva tarkkaavaisuus, eli tietoinen keskittyminen opeteltavaan asiaan. Automaattisella tarkkaavaisuudella tarkoitetaan sitä, kun ihminen



automaattisesti kiinnittää huomionsa johonkin tuttuun, tärkeään tai mielenkiintoiseen asiaan. Tähän vaikuttaa siis vahvasti käyttäjän tausta, ennakkokäsitykset ja aikaisempi kokemus järjestelmästä. Suuntautumisrefleksissä kyse on huomion kiinnittymisestä huomiota herättäviin tai uusiin kohteisiin, kuten liikeanimaatioon tai jos esimerkiksi tutun internetsivuston jokin osa näyttää erilaiselta kuin ennen. (Ketamo ym. 2004, 91–92.)

Jos mikään näytöllä ei kiinnitä erityistä huomiota, suuntaa käyttäjä katseensa yleensä ruudun vasempaan yläneljännekseen. Suunnittelussa tulisi huomioida, että ihminen havaitsee tarkasti vain pienen alueen näytöltä, eikä katseen siirtyessä välity informaatiota. Koska tarkasti nähtävä alue on pieni, ei näytön reunoille tulisi sijoittaa tärkeää sisältöä, koska se jää helposti huomaamatta. Toisaalta liike kiinnittää ihmisen huomion, vaikka se olisi aivan näkökentän reunalla. (Ketamo ym. 2004, 92.)

Kun ihminen katsoo näyttöä, hän ensiksi poimii taustasta objektit ja sitten tunnistaa ne. Käyttöliittymää suunniteltaessa tulisi välttää sellaisen käyttöliittymän tekemistä, joka hahmottuu luonnostaan eri tavalla, kuin on suunniteltu tai haluttu. Oletuksenahan on, että käyttöliittymä antaa mielikuvan siitä, millä logiikalla soveluksen toiminnallisuus eli käytännössä koodi toimii. Tämän välttämiseksi voidaan käyttää hahmolakeja. Ne kuvaavat ihmisten luontaista tapaa yhdistellä havaittuja piirteitä. Hahmolakeja on viisi, ne ovat läheisyys, samankaltaisuus, sulkeutuvuus ja yhteenliittymisen laki. (Ketamo ym. 2004, 92.)

Läheisyys tarkoittaa sitä, että lähekkäin sijoitetut kohteet nähdään ryhmänä. Tätä kannattaa hyödyntää niin, että toisiinsa liittyvät kentät, painikkeet ja säätimet sijoitellaan selvästi omaksi ryhmäksi. Näin niistä tulee selkä, oma kokonaisuutensa. Samankaltaisuus on pitkälti sama asia, kuin läheisyys, mutta liittyy ulkonäköön. Samannäköisten kohteiden mielletään liittyvän jotenkin toisiinsa. Sulkeutuvuus viittaa siihen, että visuaalisesti rajattu alue mielletään kokonaisuudeksi tai ryhmäksi. Tämä voi tarkoittaa esimerkiksi rajaviivalla tai taustavärillä rajaamista, tai esimerkiksi uuteen ikkunaan avautuva kokonaisuus on tehokkaasti rajattu muusta

sisällöstä. Yhteenliittymälaki puolestaan viittaa siihen, että toisissaan kiinni olevat kohteet liittyvät toisiinsa. (Ketamo ym. 2004, 92.)

Koska kaikkea ei hyvästä suunnittelusta huolimatta osata ottaa huomioon, tulee tuotteen käytettävyys arvioida erikseen. Tähän on kaksi lähestymistapaa, käytettävyyden tarkastus ja käytettävyydestaus. Tarkastuksessa kyse on multimedia tuotteen läpikäymisestä, jonka tarkoituksena on havaita vikoja ja puutteita. Tämä voidaan suorittaa esimerkiksi heuristisen arvioinnin avulla. Siinä tuote käydään läpi kysymyslistan avulla. Lopputuloksena saadaan lista käytettävyysongelmista ja niiden vakavuudesta. Tämän arviointikeinon huono puoli on siinä, että arvioija arvioi tuotetta oman tietonsa ja kokemuksensa perusteella, joka ei vastaa todellista loppukäyttäjää. Nielsenin tarkistuslista (Liite 1.) on hyvä esimerkki käytettävyyden arvioinnissa käytettävistä työkaluista. Siinä käydään järjestelmä läpi kymmenkohtaisen listan avulla. Arvioin omaa työtäni Nielsenin listan avulla luvussa 4. (Välinen 2007.)

Käytettävyydestauksessa tuotteen oletetun käyttäjäkunnan edustaja laitetaan käyttämään järjestelmää ja häntä seurataan tarkasti, nauhoittaen hänen puheensa, liikkeensä, sekä tietokoneen näytöllä oleva kuva. Tämän keinon ainoana huonona puolena on se, että tilanne ei täysin vastaa normaalia käyttötilannetta. Paras tulos saadaan, kun käytetään sekä käytettävyydestarkastusta, että käytettävyydestausta tuotteen arviointiin. Käytettävyydestarkastusta voidaan käyttää jo prototyyppivaiheessa, kun taas käytettävyydestausta vasta lähempänä tuotteen valmistumista. (Ketamo ym. 2004, 97–99.)

## **2.6 Multimedia opetusmateriaalin tuottamisessa**

Mitä pitää ottaa huomioon, kun käytetään multimediaa opiskelumateriaalin tuottamisessa? Mitkä ovat sen hyvä ja huonot puolet? Milloin sitä kannattaa käyttää?

Multimedian avulla opiskelu on yleensä itsenäistä opiskelua, joten lähin vertailukohta tai vaihtoehto multimedian avulla opiskelulle on yleensä lukeminen. Ryhmässä opiskelu ja lähiopetus pitävät sisällään elementtejä, joita itsenäisessä opiskelussa ei ole, oli se sitten perinteistä tai multimedian avulla tapahtuvaa. Multi-

medialla saadaan esimerkiksi lukemiseen verrattuna selvästi paremmin havainnollistettua opetettavaa asiaa kuin pelkän tekstin avulla. Opiskelusta tulee multimedian avulla visualisempaa ja auditiivisempaa sekä interaktiivisempaa. Multimedia tarjoaa monipuolisemmat työvälineet ja enemmän vaihtoehtoja kuin pelkkä teksti tai muutama kuva.

Multimediaa on erittäin helppo jakaa opiskelijoille, joilla nykyisin on lähes poikkeuksetta tarvittavat taidot, laitteet ja tietoliikenneyhteydet materiaalin käyttämiseen. Opiskelijoille voidaan tarjota internetin välityksellä opiskelumateriaalia, jota on helppo ja nopea lisätä ja muokata ilman suuria kustannuksia tai valtavaa paperimäärän jakamista. Opiskelija pääsee tähän materiaaliin myös käsiksi selvästi joustavammin, koska hän voi käyttää esimerkiksi koulun virtuaalista opiskeluympäristöä miltä tahansa tietokoneelta, jossa on internetyhteys. Näin ollen hän ei ole sidottuna koulurakennukseen kuten perinteisessä opetuksessa.

Olen huomannut, että esimerkiksi opetusvideot ovat selvästi tehokkaampi ja mielekkäämpi tapa opiskella esimerkiksi jotain tietokoneohjelmaan liittyvää aihetta, kuin samasta aiheesta kirjoitettu teksti. Virheitä ja väärinymmärryksiä tulee selvästi vähemmän, kun käyttäjä näkee mitä videolla tapahtuu ja tehdään sen sijaan, että luetaan pitkä sanallinen kuvaus aiheesta. Olen käyttänyt opetusvideoita apuna 3D-grafiikan opiskelussa ja huomasin, että jaksoin opiskella aihetta selvästi pidempään videoiden avulla kuin kirjoitettujen ohjeiden avulla. Video on monesti selvästi havainnollisempi keino opettaa, kuin pelkkä teksti. Opin myös samassa ajassa enemmän ja väärinymmärryksiä ja virheitä tuli vähemmän, koska puhujan äänen lisäksi näin videolta, mitä hän ohjelmassa teki. Jos samat asiat kuvaillaan tekstillä, tulee tekstiä niin paljon, että sen lukeminen ja ymmärtäminen kestää selvästi kauemmin kuin videon katsominen. Puhumattakaan väärinkäsityksistä, jotka johtuvat ihmisten erilaisesta tavasta ilmaista asioita kielellisesti.

Verkko-opiskelijoiden oppimistyyli painottuu aktiiviseen ja visuaaliseen suuntaan, muihin opiskelunmuotoihin verrattaessa (Vainionpää 2006, 192). Nykyopiskelijalle, joka on tottunut käyttämään internetiä, pelaamaan pelejä ja katsomaan videoita tietokoneellaan, on multimedian käyttäminen opiskeluvälineenä varsin

luontevaa. Tuottamani videot painottavat verkko-opiskelun visuaalista puolta ja opiskelijoiden tehtävien teko aktiivista.

Huonoja puolia multimedian käyttämisessä on se, että kaikilta opiskelijoita ei välttämättä löydy tarvittavia ohjelmia, niiden oikeita versioita tai oikeita koodekkeja jne. Myös tekniset ongelmat voivat aiheuttaa hankaluuksia verkko-opiskelussa. Multimediaa kannatta käyttää silloin, kun halutaan tehdä itsenäisestä opiskelusta interaktiivista tai halutaan opettaa jostain syystä jokin kurssi lähiope- tuksen sijaan verkossa.

### **3 OPETUSMATERIAALIN TUOTTAMINEN**

Tässä luvussa käsittelen työni tekemistä sekä suunnittelua. Kerron myös mihin pyrin työtä tehdessäni ja miksi, sekä perustelen ratkaisuja, joihin olen päätenyt.

#### **3.1 Tehtävien toteutus**

Hain internetistä inspiraatiota mahdollisiin tehtäviin, sekä koodi vinkkejä erilaisilta ActionScript 3-aiheisilta sivustoilta. Yksi tehtävä, joka minun tulisi tehdä, oli ennalta määrätty. Muut tehtävät saisin suunnitella itse. Ennalta määrätty tehtävä oli tehtävä numero 11. Suunnitteluvaiheessa pyrin siihen, että tehtävien sisältö vaihtelisi, ja että jokaisessa olisi jotain uutta opittavaa ActionScript 3-ohjelmoinnista. Tavoitteena oli tehdä sellaisia tehtäviä, joiden tekeminen olisi mielekästä ja mahdollisimman mielenkiintoista opiskelijoille. Siksi tehtävien joukossa on esimerkiksi galleria, MP3-soitin ja pari peli.

Tein tehtävät varsin joustavasti. Sitä mukaa, kun sain tehtäviä tehdyksi, selkenivät suunnitelmani siitä, millaisia jäljellä olevien tehtävien tulisi olla. Ensimmäistä tehtävää tehdessäni en vielä osannut muodostaa kokonaiskuvaa kaikista lopuista tehtävistä, mutta mitä enemmän olin ohjelmoinut ja tutkinut erilaisia sovelluksia ja opetusmateriaalia, sitä helpommin minulle muodostui mielikuva siitä, mikä on sopiva sisältö tälle kurssille ja millaisia mielekkäitä vaihtoehtoja tehtävien aiheiksi on.

Tehtävissä käytetään useita kuvia, yhtä MP3-tiedostoa, sekä ulkoista tekstitiedostoa. Tekstitiedoston opiskelijat voivat helposti luoda itse, mutta kuvat ja MP3:n tarjoan valmiina liitetiedostona kunkin tehtävän yhteydessä. Kaikki kuvat ovat omiani ja MP3-tiedosto on Teosto-vapaa ja minulla on lupa käyttää sitä tähän tarkoitukseen.

#### **3.2 Videoiden suunnittelu ja toteutus**

Päätettyäni tehtävän sisällön tein sen kokonaisuudessaan valmiiksi. Tämän jälkeen tein jokaisesta tehtävästä oman käsikirjoituksensa, jota käytin apuna tehdessäni tehtävät uudelleen, samalla tekoprosessin nauhoittaen. Tavoitteena oli videota

kaapatessa tehdä tehtävä kerralla läpi. Tähän käytin apuna käsikirjoitusta, johon olin merkinnyt mitä teen missäkin vaiheessa sekä kirjoitettavan koodin. Tätä käsikirjoitusta pidin toisella näytöllä, ja toisella näytöllä oli auki Flash. Kaappasin vain sen näytön kuvan, jolla Flash oli auki.

Videoiden käsikirjoitus perustui siis täysin tehtävän vaiheisiin. Vaiheet olivat karkeasti sanottuna aina samat. Ensin loin tarvittavat tasot, kehykset ja objektit ja asettelin objektit oikeisiin kehyksiin ja oikeille tasoille. Kun kaikki aikajanalla oli valmista, aloitin ohjelmoinnin. Lopuksi kokeilin, että sovellus toimii ja esittelin tehdyn työn toiminnallisuudet. Eniten aikaa vievissä tehtävissä tein kaksi erillistä videota, toisessa sisällön luominen aikajanalle ja toisessa koodin kirjoittaminen.

Videon kaappaamisen jälkeen tein jälkiäänityksenä kommentit videolle. Kommenttien tarkoituksena on kertoa, mitä videolla tehdään ja miksi. Äänittäminen videota kaapatessa olisi ollut mahdollista, mutta aivan liian haastavaa, sillä ohjelmaan, käsikirjoitukseen ja puhumiseen keskittyminen samaan aikaan, ilman virheitä olisi ollut lähes mahdotonta. Tästä syystä päädyin tekemään äänityksen jälkikäteen. Näin pystyin tekemään äänityksen osissa ja äänittämään jonkin kohdan uudestaan, jos se epäonnistui ensimmäisellä kerralla.

Käytin videoiden kaappaamiseen FlashBack Pro Recorderia ja editointiin, äänittämiseen ja pakkaamiseen FlashBack Pro Playeria. Pystyin tekemään kaikki tarvittavat toimenpiteet näillä kahdella ohjelmalla. Recorder loi .fbr-tiedoston, jota sitten Palyerilla pystyin muokkaamaan. Kokeilin useita eri ohjelmistoja, mutta päädyin tähän, koska sillä pystyin tekemään kaiken, ilman että minun olisi tarvinnut pakata video ja jatkaa sen muokkaamista toisessa ohjelmassa ja pakata se taas uudestaan. Muissa kokeilemissani ohjelmissa oli jotain puutteita, tai en saanut niitä toimimaan kunnolla.

Sekä FlashBack Pro recorder, että Player olivat molemmat erittäin onnistuneita käytettävyydeltään. Ilman minkäänlaista kokemusta tai ohjeistuksen käyttöä pystyin nauhoittamaan, leikkaamaan ja äänittämään videot. Käyttöliittymä on erittäin selkeä ja tehokas. Esimerkkinä vaikkapa se, että ohjelma tallentaa kaikki videon teon aikana tapahtuvat näppäimen painallukset. Kun videota sitten editoidaan, voi

käyttäjä hakea halutun kohdan pelkän etsimisen sijaan vaikkapa avaamalla ikkunan, jossa näkyy lista komentoista, joita on käytetty ja painamalla jotain niistä. Ohjelma vie käyttäjän siihen kohtaan, jossa kyseinen painallus on tapahtunut. Näppäinkomennon näkyvät myös suoraan aikajanalla.

### 3.3 Opetusmateriaalin jakaminen verkossa

Koska työn tilaaja, eli Vaasan ammattikorkeakoulu käyttää internet-pohjaista virtuaalista opiskeluympäristöä, Moodlea, oli luonnollinen valinta sijoittaa myös minun työni sinne. Ei olisi ollut johdonmukaista sijoittaa yhden kurssin materiaaleja jonnekin erilliselle, tarkoitusta varten tehdyille sivustolle, koska opiskelijat käyttävät jo muiden kurssin yhteydessä Moodlea. Kurssin nimi Moodlessa on Flash ActionScript 3 -kurssi.

Varsinaiset opetusvideot ovat YouTubessa sen joustavuuden takia. Suoraan Moodleen sivulle ei voi upottaa videoita, joten erillinen tai ulkopuolinen sivu videoille olisi joka tapauksessa pitänyt tehdä. YouTube tukee lähes kaikkia videotiedostotyyppisiä ja katsoja voi itse valita, minkä kokoisena ja laatuksena hän katsoo videon. YouTubeen on myös erittäin helppo ja nopea lisätä videoita ja muokata niiden tietoja. Näistä syistä päädyin YouTubeen käyttämiseen. Loin YouTubeen uuden käyttäjätunnuksen, jota käyttäisin vain näiden opetusvideoiden jakamiseen.

Kaikki muu sisältö, eli tehtävien kuvaukset, koodi kuvatiedostoina, sekä tarvittavat liitetiedostot ovat Moodlessa. Sivun yläreunassa, ennen tehtäviä on kuvaus kurssista, sen sisällöstä ja tarpeelliset ohjeet. Jokainen tehtävä on oma ”osionsa” Moodle-kurssin sivulla. Kukin tehtävä osio alkaa kuvauksella tehtävän sisällöstä ja siitä, mitä uutta kyseisessä tehtävässä opitaan. Tämän jälkeen on linkki tehtävässä tarvittavaan oheismateriaaliin, jos sellaista on. Seuraavaksi tulee linkki tehtävää vastaavaan YouTube-videoon sekä viimeisenä linkki uudessa ikkunassa aukeavaan kuvatiedostoon, jossa näkyy tehtävän koodi kokonaisuudessaan. Kurssia käyttävä opettaja voi halutessaan piilottaa linkit, joiden takaa näkee koodin, jos hän haluaa, että opiskelijat varmasti katsovat ja kuuntelevat videolta koodin teon. Näin he oppivat varmemmin koodin merkityksen ja toiminnallisuuden.

## 4 TUOTETTU VERKKO-OPETUSMATERIAALI

Tässä osiossa käydään läpi jokaisen tekemäni opetusvideon sisällön teema ja toiminnallisuus, sekä se mitä uutta kyseisessä tehtävässä on opiskelijalle. Tehtävien sisältämä koodi löytyy liitteestä numero 1.

Kuvassa 1 näkyy moodle-sivun sisältöosan alku, eli alkuteksti ja ensimmäinen tehtävä. Kussakin tehtävässä on ensin otsikko, sitten kirjoitettu kuvaus tehtävästä. Tämä kuvaus on sama kuin tässä osiossa myöhemmin tulevat kuvaukset. Viimeisenä tulevat linkit opetusvideoon, koodiin, sekä jos tehtävässä tarvitaan ulkoisia tiedostoja, myös niihin (tehtävät 7, 8, 9, 10).

### Flash ActionScript 3 -kurssi

Tervetuloa Flash ActionScript 3 -kurssille. Kurssin tavoitteena on opettaa Flash ActionScript 3 -ohjelmointikieltä etäopiskelun keinoin.

**Käyttöohje**



Tarkoitus on, että tehtävät tehdään järjestyksessä. Tehtävät muuttuvat vaikeammiksi loppua kohden.

Lue ensin tehtävän kuvaus. Tehtävissä 7, 8, 9 ja 10 tarvitsen tehtävän tekemiseen tiedostoja, jotka löytyvät liitteenä kyseisten tehtävien koodista. Näiden tehtävien tapauksessa aloita luomalla kansio, johon tallennat kyseisen tehtävän tekemiseen tarvittavat tiedostot. Kun teet itse Flash-tehtävää, tallenna se samaan kansioon, johon tallensit kyseiseen tehtävään liittyvät tiedostot. Tehtävässä 12 koodi tallennetaan erilliseen tiedostoon ja tämän tiedoston tulee olla samassa kansiossa kuin Flash-tiedoston, joten senkin tapauksessa suosittelen erillisen kansion luomista tehtävää varten.

Muiden tehtävien tapauksessa ei ulkoisia tiedostoja käytetä, mutta suosittelen silti, että teette niillekin omat kansionsa, sillä jokaisesta tehtävästä tulee ainakin .fla ja .swf tiedostot. Joten on selkeämpää tehdä jokaiselle tehtävälle oma kansionsa.

Katso opetusvideo ja tee tehtävä sen pohjalta. Videot ovat YouTubessa. Kunkin tehtävän kohdalla on linkki joka vie tehtävää vastaavaan videoon. Muista laittaa YouTubessa HD-laatu ja fullscreen-toiminto päälle.

Suosittelen videon katsomista osissa ja työn tekemistä niin, että video on pysäytettynä. Kun tehtävä on valmis ja toimii halutulla tavalla, se palautetaan palautuslinkin kautta.




 youtube asetukset (kuva)  
 Uutiset

1

### 1. tehtävä, painike

Ensimmäisessä harjoituksessa luodaan painike, jota painettaessa dynaamiseen tekstikenttään tulee tieto siitä, montako kertaa painiketta on painettu. Tässä tehtävässä opitaan tekemään muuttuja, luomaan event listener, sekä funktio. Kirjoitamme myös tekstiä stagella olevaan tekstikenttään.

Linkki tutorial-videoon. Muistakaa käyttää youtubessa tarjolla olevaa HD-vaihtoehtoa ja full screen -tilaa.

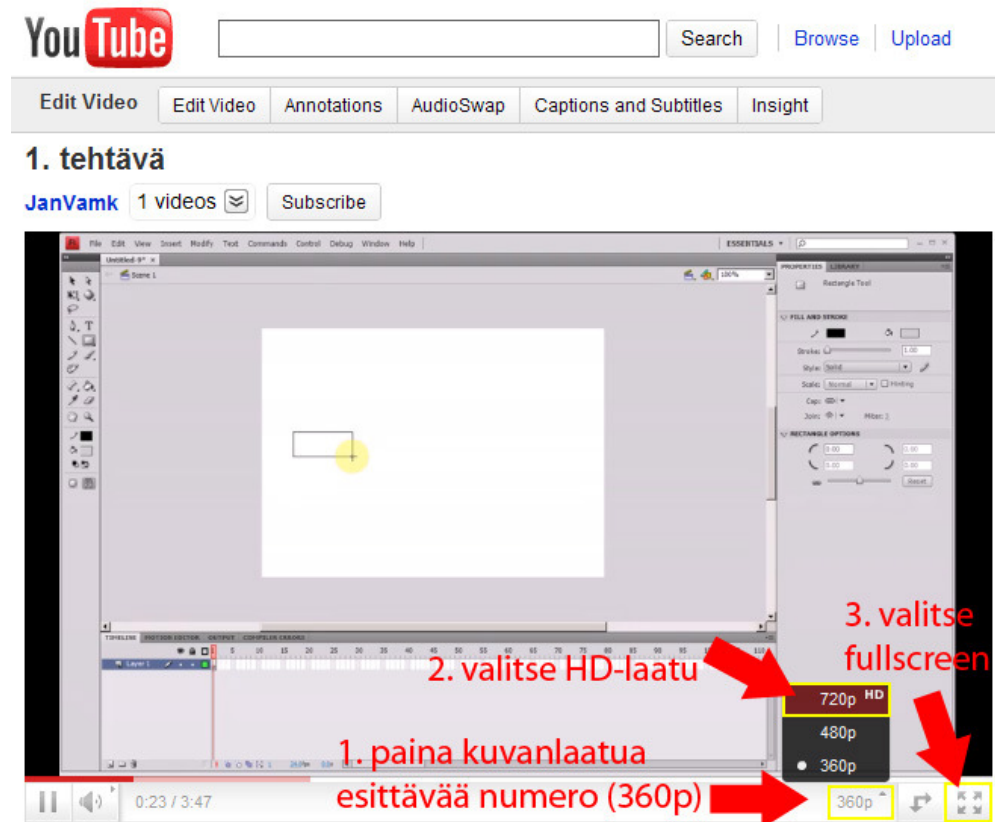
-  [linkki tutorial videoon](#)
-  [koodi](#)
-  [1. tehtävä painike palautus](#)

Kuva 1. Moodle-sivusto

Alkutekstin jälkeen on linkki, josta aukeaa uuteen ikkunaan kuva, jossa on opastettu miten YouTubesta saadaan oikeat asetukset päälle, jotta opetusvideon katsominen olisi mahdollisimman helppoa ja käyttäjäystävällistä (kts. Kuva 2). Jotta



ohje olisi mahdollisimman selkeä, käytin huomioväreinä punaista ja keltaista, sekä laitoin toimenpiteet numerojärjestykseen.



Kuva 2. YouTube-ohje

Seuraavaksi käydään läpi jokaisen tehtävän toiminnallisuus sekä tehtävässä opittavat asiat. Tehtävien koodi löytyy kokonaisuudessaan liitteestä numero 1.

### Harjoitus 1, painike

Ensimmäisessä harjoituksessa luodaan painike, jota painettaessa dynaamiseen tekstikenttään tulee tieto siitä, montako kertaa painiketta on painettu. Tässä tehtävässä opitaan tekemään muuttuja, luomaan event listener sekä funktio. Kirjoitamme myös tekstiä näyttämöllä olevaan tekstikenttään. (Liite 2.)

## **Harjoitus 2, netti nappula**

Toisessa tehtävässä luodaan input-kenttä, johon käyttäjä voi kirjoittaa www-osoitteen, ja painiketta painettaessa avautuu selainikkuna, joka lataa käyttäjän kirjoittamaa osoitetta vastaavan sivun. Tässä tehtävässä opetellaan URLRequestin käyttöä. URLRequestilla voidaan viitata koneen hakemistossa olevaan tiedostoon tai www-osoitteeseen. (Liite 2.)

## **Harjoitus 3, teksti ulkoisesta tiedostosta**

Kolmannessa tehtävässä tehdään sovellus, jossa nappia painamalla saadaan tuotua teksti ulkoisesta tekstitiedostosta dynaamiseen tekstikenttään. Tehtävässä opetellaan hakemaan tiedosto koneelta URLRequestin avulla ja ladataan se URLLoaderilla sovellukseen. (Liite 2.)

## **Harjoitus 4, laskin**

Tässä tehtävässä tehdään laskin. Käyttäjä syöttää kahteen input-tyyppiseen tekstikenttään kaksi numeroa, joilla voidaan tehdä, plus, miinus, kerto ja jako -laskut. Tässä tehtävässä käsitellään boolean muuttujia, tekstitiedon muuttamista numerotiedoksi ja takaisin tekstiksi sekä tekstikenttien ominaisuuksien hyödyntämistä. (Liite 2.)

## **Harjoitus 5, digitaalinen kello**

Viidennen tehtävän teemana on digitaalinen kello. Tarkoituksena on tehdä kello, jossa näkyy kellonaika numeroina, sekä päivämäärä niin, että kuukausi on sanamuodossa numeron sijaan. Tässä tehtävässä opitaan date-luokan käyttö. Tietokoneen kellolta haetaan päivämäärä- ja kellonaika-tiedot ja ne viedään dynaamisiin tekstikenttiin, käyttäjän nähtäville. (Liite 2.)

## **Harjoitus 6, Flash sovellus Flash sovelluksessa**

Tässä tehtävässä tehdään flash-sovellus, johon avataan kaksi toista Flash-sovellusta. Tässä tapauksessa avaamme tämän sovelluksen sisällä jo tehdyt laskin- ja kello -sovellukset. Tässä tehtävässä opitaan siis avaamaan Flash-sovellus Flash-

sovelluksen sisällä. Tämän lisäksi käydään läpi avattavien sovellusten sijoittaminen Flash-esitykseen x ja y-koordinaattien avulla. (Liite 2.)

### **Harjoitus 7, galleria**

Luodaan galleria, jossa on viisi pikkukuvaa. Yksi niistä on oletuksena valittuna, kun sovellus avataan. Valittuna oleva kuva näkyy isompana pikkukuvien vieressä. Kun käyttäjä painaa jotain muuta kuin valittuna olevan kuvan pikkukuvaa, vaihtuu iso kuva täksi kuvaksi. Kuvat ladataan ulkoisesta tiedostosta. Tässä tehtävässä opitaan hakemaan ja lataamaan ulkoisia kuvia näyttämölle sekä käsittelemään movie clippejä kuten painikkeita. (Liite 2.)

### **Harjoitus 8, preloader**

Kahdeksannessa tehtävässä ladataan isohko kuvatiedosto ennen varsinaisen sovelluksen alkamista. Eli ennen kuin mennään kehykseen, jossa kuva tulisi näkymään, näytetään latauksen edistyminen prosentteina ja progress barin avulla ja vasta latauksen valmistuttua siirrytään kehykseen, johon kuva on latautunut. Uutta tässä tehtävässä on loader info-objektia käyttäminen latauksen edistymisen esittämiseen. (Liite 2.)

### **Harjoitus 9, palapeli**

Alku tilanteessa käyttäjä näkee palat epäjärjestyksessä vasemmalla ja oikealla harmaan alueen, jolle kuvat pitäisi laittaa oikeaan paikkaan niin, että ne muodostavat kuvan. Palapelin palat ovat liikuteltavia movie clip –objecteja. Käyttäjä voi ottaa niistä hiirellä kiinni, painamalla hiiren oikeaa painiketta ja raahata palaa ympäri näyttämöä. Jos hän vie sen oikeaan kohtaan harmaalle taustalle ja päästää palasta irti, se napsahtaa paikoilleen, eikä sitä voi enää liikuttaa tämän jälkeen. Jos käyttäjä päästää palasta irti muualla kuin oikeassa kohdassa, palaa pala automaattisesti lähtöpisteeseensä. Kun palapeli on valmis, tulee onnitteluteksti näkyviin. Tässä tehtävässä opitaan hyödyntämään painike ominaisuutta movie clipin kanssa ja raahaustoimintoa. (Liite 2.)

### **Harjoitus 10, MP3-soitin**

Tehdään soitin, joka soittaa tietokoneella olevan MP3:n. Käyttäjälle tarjotaan play- ja stop-painikkeet, sekä volume-slaideri, jolla hän voi säätää äänen voimakkuutta. Tehtävässä opetellaan Sound, SoundChannel, sekä SoundTransform -luokkien käyttöä. (Liite 2.)

### **Harjoitus 11, Pertin pelit**

Tässä tehtävässä tehdään nopeuspeli. Käyttäjä näkee kolme eri väristä ympyrää, punaisen, sinisen ja vihreän. Yksi näistä ympyröistä on aina keltainen lyhyen aikaa. Käyttäjän on tarkoitus painaa keltaisen olevaa ympyrää. Se, mikä ympyrä on keltainen kunakin hetkenä, on täysin sattumanvaraista. Jos pelaaja painaa keltaista ympyrää, hän saa yhden pisteen. Jos hän painaa muun väristä ympyrää, hän saa virhe pisteen. Peli päättyy, kun pelaajalla on viisi virhe-pistettä. Pelaajan pisteet ja virhepisteet näytetään hänelle jatkuvasti pelin aikana dynaamisissa tekstikentissä. Kun peli on ohi, peli pysähtyy ja näkyviin tulevat tekstit "Game over!" sekä "Pisteesi ovat:". Tässä tehtävässä opitaan satunnaisluvun luominen ja hyödyntäminen. (Liite 2.)

### **Harjoitus 12, tasohyppely**

Viimeisessä tehtävässä tehdään peli, jossa pelaaja liikuttaa punaista neliötä, eli "pelaajaa". Pelaajan tulee hyppiä nuolinäppäimiä liikkumiseen käyttäen tasolta toiselle. Jos hän ei osu tasolle, vaan putoaa ohi ja pois näkyvältä alueelta, peli alkaa uudelleen alusta. Pelaaja pysyy koko ajan keskellä näyttämöä, ympärillä olevat tasot liikkuvat, joka saa pelin näyttämään siltä, että pelaaja liikkuisi. Tehtävässä käydään läpi ulkoisen AS3 -tiedoston käyttäminen, näppäimistön painikkeiden painamisen "kuunteleminen" ja hyödyntäminen, sekä hitTest -metodi.

Peli toimii, mutta ei ole valmis vielä tämän opetusvideon jälkeen jälkeen. Siinä on tiettyjä puutteita, joita opiskelija voi halunsa ja taitojensa mukaan korjata omaisesta. (Liite 2.)

Käyttäjäpalautteen puutteen vuoksi testasin videoita Nielsenin 10 kohdan-arvioinnin avulla. Käytin sovellettua versiota, johon olen valinnut 6 parhaiten työtäni arvioivaa kohtaa. Käytin kohtia 2, 3, 4, 7, 8 ja 9 (Liite 2.). (Välinen 2007.)

Videoiden ja todellisen maailman välinen yhteys on niin suora kuin mahdollista. Käyttäjä näkee videota ohjeiden mukaan katsoessaan ruudunkokoisena kuvan siitä, mitä tehtävässä tulee tehdä. Käyttäjän hallinta ja vapaus toteutuvat YouTubeen tarjoamissa puitteissa. Käyttäjä voi valita äänen voimakkuuden, ruudun koon ja kuvan tarkkuuden, YouTubeen tarjoamista vaihtoehtoista. Käyttäjä voi pysäyttää videon tai siirtyä aikajanalla haluamaansa kohtaan. Videoiden yhtenäisyys teknisesti ja käsikirjoituksellisesti on erimainen. Kuitenkin epäyhtenäisyyttä tulee videoiden sisällössä. Nopeus, jolla kirjoitan koodia sekä teen muita tehtävän vaiheita vaihtelee. Myös puhenopeuteni ja käyttämäni puheäänien voimakkuus vaihtelevat. Tämä saattaa aiheuttaa käyttäjälle vaikeuksia. Käytön tehokkuus ja joustavuus ovat varsin hyvällä tasolla. Käyttäjä voi katsoa videoita silloin, kun hän haluaa. Hän voi valita missä järjestyksessä katsoo videoita ja hän voi katsoa videosta haluamansa kohdat tai siirtyä eteenpäin. YouTube-videot ovat suoratoisto teknikalla toteutettuja, eli niitä voi katsoa haluamastaan kohdasta ilman, että koko video on ladattu. Tämä säästää käyttäjän aikaa. Käyttöliittymän estetiikka ja malli ovat YouTubeen omia. Itse video on äänenlaadultaan moitteetonta. Video voisi olla aavistuksen tarkempaa, mutta lähinnä esteettisistä syistä. Käytön kannalta kuvan tarkkuus on riittävä. Videoiden sisältöön liittyvä ongelman käsittely liittyy lähinnä siihen, jos käyttäjä ei saa itse tekemäänsä sovellusta toimimaan, kuten se opetusvideolla toimii. Tässä tilanteessa hän voi käyttää avukseen Moodlesta löytyvää erillistä tiedostoa, jossa on tehtävän koodi. Hän voi käyttää tätä koodia tarkistaakseen tekemänsä koodin. Moodlesta löytyy myös lista yleisistä virheistä, joita käyttäjä saattaa tehdä tehtäviä tehdessään. Näiden lisäksi muuta apua tehtävien teossa ilmeneviin ongelmiin ei ole. Koska erilaisia ilmeneviä ongelmia on lukuisia, olisi ainoa kaiken kattava ratkaisu se, että käyttäjä voisi kysyä apua. Verko-opiskeluformaatti ei tarjoa tähän ongelmaan helppoja ratkaisuja. Puhtaasti tekniset ongelmat videoiden kanssa liittyvät YouTube palveluun.

## 5 YHTEENVETO

Tässä osiossa arvioin ja käyn läpi työ prosessia sekä lopputulosta. Pohdin myös tulosten merkitystä, luotettavuutta, sovellusmahdollisuuksia, sekä mahdollisia jatkotutkimustarpeita.

### 5.1 Prosessin kuvaus ja arviointi

Toimeksiannossa annetut tavoitteet sisällön tuoton osalta täyttyivät mielestäni erittäin hyvin. Opetusvideot sain toteutettua haluamallani ja tavoitteeksi asetetulla tavalla. Videoiden kuvan laatu olisi voinut olla aavistuksen verran parempi, mutta en löytänyt sellaista tiedostotyyppiä, koon ja koodekin yhdistelmää, jolla kuva olisi YouTubeen viemisen jälkeenkin ollut täysin terävä. Kuvan laatu on kuitenkin täysin riittävä, sillä mitään niin pientä yksityiskohtaa ei videolla ole, ettei siitä saisi selvää. Heikon kuvanlaadun kanssa ensimmäiseksi ongelmia aiheuttaisi tekstin lukeminen, mutta tätä ongelmaa ei videoideni kanssa ole. Videoille jälkiäänitetty puheeni on todennäköisesti videoiden laadultaan epätasaisin osa-alue. Olen pyrkinyt tekemään harjoituksista selkeitä ja virheettömiä. Varsinaista käyttäjätestausta ei ollut mahdollista tehdä, joten puheideni ”käyttäjystävällisyydestä” ei ole tutkittua tietoa.

Moodle-kurssista tuli sellainen kuin oli tarkoituskin, niin ulkoasultaan, kuin sisällöltään. Kaiken kaikkiaan olen käytännön työhön tyytyväinen. Sain tehtyä sovitun määrän opetusvideoita, joiden sisältö on monipuolinen ja mielekäs. Käyttäjälautteen puutteen vuoksi videoiden selkeydestä ei ole täyttä varmuutta.

Työn tekeminen oli melko positiivinen kokemus. Minulla ei ollut missään vaiheessa suurempia vaikeuksia työn edistymisen suhteen. Flash-tehtävien kanssa oli välillä ongelmatilanteita, mutta jatkoin aina jonkin toisen tehtävän tekemistä tällaisissa tilanteissa. Tämä osoittautui oikeaksi ratkaisuksi. Saatoinkin keksiä ratkaisun ongelmalliseen tehtävään tehdessäni jotain toista tehtävää, esimerkiksi sitä varten tietoa hakiessani. Tosin jossain tapauksessa jouduin aloittamaan tehtävän lähes puhtaalta pöydältä, jos vikaa ei löytynyt. Myös sopivan tiedostomuodon, koodekin

ja muiden videoasetusten löytäminen videon ulostuonti (export) vaiheessa vaati melkoisesti kokeilemista. Muita selkeitä vaikeuksia tai hidasteita ei ollut.

Koin itse varsin onnistuneeksi työjärjestyksen jota käytin. Hain perustietoa alussa ja kirjoitin teoriaosaa työstäni. Tämän jälkeen aloin tehdä käytännön työtä, jonka jälkeen kirjoitin itse raporttiosan työstä. Raportointi ja yleensäkin Flashista tai ActionScriptistä minun oli helpompi kirjoittaa vasta käytännöntyön jälkeen, kun taas perustietojen hakeminen alussa oli ihan hyödyllistä, jotta sain vähän perustietoa työni tueksi ja sain tuntuman kirjoittamiseen ja sen aikatauluttamiseen. Myös se osoittautui oikeaksi ratkaisuksi, että en heti päättänyt väkisin, millaisia Flash harjoituksia tekisin, vaan päätin muutaman ja aloitin niiden tekemisen. Tehtäviä tehdessä ja tietojeni ja näkemykseni karttuessa osasin paremmin päättää, millaisia loppujen tehtävien kannattaisi olla. Eli joskus liika suunnitteleminenkin voi olla haitaksi ja tuhлата aikaa. Luonnollisesti tämä ei päde moneen muuhun opinnäytetyön osa-alueeseen.

Jos tekisin työni uudelleen tai jos aikaa olisi selvästi enemmän, pyrkisin testaamaan videot käyttäjillä. Näin saataisiin palautetta videoiden selkeydestä ja mahdollisista puutteista. Puutteella tarkoitan tässä tapauksessa tilannetta, jossa videolla ei selitetä, mitä tehdään tai miksi tehdään. Tietysti myös asiavirheet ovat mahdollisia. Videoiden editoinnin seurauksena on teoriassa mahdollista myös se, että jotain jää vahingossa näyttämättä ja näin käyttäjä ei saa sovellustaan toimimaan vaikka seuraisikin ohjeita. Parhaaseen tulokseen päästäisiin todennäköisesti niin, että jokainen opetusvideo annettaisiin mahdollisimman monen käyttäjän katsottavaksi niin, että he tekevät tehtävän itse videon ohjeiden pohjalta. Testaajia pyydetäisiin raportoimaan kaikki puutteet ja virheet videoissa.

Kirjalliset lähteet valitsin täysin sen perusteella, mitkä teokset vastasivat aiheeltaan parhaiten työtäni. Valitsemani kirjat vastasivat parhaiten hakuun laittamiani hakusanoja. Haettuani kirjat huomasin, että ne vastasivat myös sisällöltään varsin hyvin sitä, mitä olin etsimässä. Jätin lähteitä käyttämättä ainoastaan, jos ne eivät mielestäni tarjonneet aiheeseen liittyvää tietoa.

Valitsemani kirjat olisivat voineet olla monipuolisempia. Päädyin käyttämään suomenkielistä kirjallisuutta ja kahdessa käyttämässäni kirjassa oli samat tekijät. Multimedia. Nyt -kirjan ovat kirjoittaneet asiantuntijat, joista mainittakoon Professori Jari Multisilta sekä Dosentti Harri Ketamo. Digitaalinen viestintä-kirjan ovat kirjoittaneet Keränen, verkkopalvelujen ohjelmistosuunnittelija sekä kouluttaja, AV-alan kouluttaja ja projektipäällikkö Penttinen sekä julkaisun ja kuvankäsittelyn ammattilainen Lamberg. Tekijät ovat valitsemisani kirjoissa melko päteviä ja siten lähteenikin luotettavia. Kirjoittajilla on yliopistokoulutus ja/tai paljon kokemusta työelämästä. Varsinaisia ristiriitoja ei eri kirjojen välillä tullut vastaan.

Koska lähteeni vaikuttivat pääasiassa teoriaosuuden sisältöön, sanoisin, että varsinaisen työn lopputulokseen lähteilläni ei ollut juurikaan vaikutusta. Sen sijaan opetusvideoilla oleviin Flash AS3-tehtäviin sain paljon vaikutteita lukemattomilta internetsivuilta, joista hain ajatuksia, inspiraatiota ja ratkaisuja ongelmiini. Internetissä näkemäni sisältö vaikutti siis jossain määrin tietoisesti ja alitajuisesti siihen, millaisia tehtävistä tuli. Esimerkiksi flashhandmath.com-sivustolta sain ideoita tehtävien sisältöön.

Opinnäytetyön luotettavuus ja onnistuminen tulee esille harjoitusten toistettavuutena ja siirrettävyytenä myös muihin verkko-opetuskokonaisuuksiin. Videoiden sisältö on varsin helppo toistaa opiskelijan toimesta, sillä kaikki tehtävän tekemiseen liittyvät vaiheet on kertomisen lisäksi näytetty videolla. Kommunikointiin liittyviä ongelmia ei pitäisi ilmetä, kuten pelkän suullisen tai kirjallisen opetuksen yhteydessä voi helposti käydä.

Opetusvideot ovat varsin tarkasti selostettuja. Esimerkiksi käytetyt näppäinoikopolut on selitetty. Jos käyttäjällä on Flash CS4 käytössään ja hän saa sen avattua, pitäisi hänen pystyä seuraamaan ja hyödyntämään videoilla olevaa opetusta riippumatta siitä, missä yhteydessä niitä käytetään. Kielirajoitus luonnollisesti on olemassa, sillä ohjeet ja videoilla oleva puhe on vain suomeksi. Itse videoita ei tarvitse siirtää mihinkään, koska ne ovat valmiiksi ulkoisessa ja julkisessa paikassa, eli YouTube:ssa. Ainoa työtä teettävä osa videoiden käytössä muun verkko-



opetuskokonaisuuden yhteydessä on liitetiedostojen ja sanallisten ohjeiden siirtäminen.

Moodle asettaa tiettyjä rajoituksia sen suhteen, miltä kurssin sisältö voi näyttää. Itse videot ovat kuvanlaadultaan hyviä ja antavat melko ammattimaisen kuvan. Sen sijaan videoille jälkiäänitetyt puheet eivät yllä samalle tasolle. Äänen käyttöni voisi olla jämäkämpää ja ryhdikkäämpää.

Näkisin, että työni on melko realistinen. Tehtävät vaikeutuvat loppua kohden ja niissä opeteltavat asiat ovat varsin käytännönläheisiä. Opetettavat asiat kattavat AS3-ohjelmoinnin perusteita sekä yleisesti tarvittavia taitoja. Pysin valitsemaan tehtäville mahdollisimman motivoivan sisällön. Tämä näkyy siten, että tehtävät ovat varsin käytännönläheisiä, kuten laskin, MP3-soitin, kello. Yritin saada tehtävistä myös viihdyttäviä, esimerkiksi sisällyttämällä kaksi peliä tehtävien joukkoon. Olisin voinut olla vielä motivoivampi videoilla olevissa puheosuuksissani. Joku saattaa mahdollisesti kokea ne hieman tasapaksuiksi tai tylsiksi, mutta toisaalta ne ovat ainakin selkeitä.

Kriittisiä resursseja toteutettavuuden suhteen ovat lähinnä aika. Mikä on kohtuullinen aika, jossa käyttäjän voidaan olettaa tekevän tehtävät. Siitä ei ole tutkittua tietoa. Tehtävien tekoaika videoiden avulla vaihtelee todennäköisesti parista kymmenestä minuutista reiluun tuntiin. Ongelmien ilmetessä voi aikaa kulua selvästi kauemmin. Tällaisia ongelmia ovat esimerkiksi vaikeasti havaittava virhe koodissa. Karkea arvio voisi olla, että käyttäjällä kestää noin 10–15 tuntia tehdä kaikki 12 tehtävää.

Tehtävien sisältö vastaa varsin hyvin työelämän vaatimuksia siitä, mitä työelämässä voidaan vaatia palkattavan henkilön osaavan. Luonnollisesti nämä 12 tehtävää eivät tarjoa riittävää koulutusta ammattimaisen AS3-ohjelmoinnin hallitsemiseen, mutta ne pitävät sisällään niitä perustaitoja, jotka ammattilaisen tulee hallita.

## 5.2 Tulokset

Toimeksianto oli melko tarkka sen suhteen, että lopputuloksena syntyisi vähintään 10 AS3-opetusvideota ja että kurssin sijoitettaisiin Moodleen. Joku toinen olisi tehnyt ainakin osittain erilaisia tehtäviä omien näkemystensä ja kokemustensa pohjalta. Tehtävien pedagogiikkaan en ota kantaa. Flash AS3:ssa opetettavia asioita on niin paljon, että vain osa niistä voidaan tämän kurssin puitteissa käydä läpi. Jokaisella on oma näkemyksensä siitä, mitkä olisivat olennaisimpia asioita. Itse laitoin videot YouTube-palveluun, joku toinen olisi saattanut laittaa videot johonkin muuhun palveluun tai koulun palvelimelle.

Koska kyseessä oli toimeksianto, joka sisälsi pääasiassa käytännöntyötä, lähestyn tutkimuskysymyksiä teorian näkökulmasta. Seuraavassa käsittelen lyhyesti kutakin tutkimuskysymystäni.

### **Multimedian mahdollisuudet opiskelumateriaalin tuottamisessa**

Multimedia on erittäin joustava tapa tarjota opiskelumahdollisuuksia. Tietoverkkojen avulla opiskelu on mahdollista käytännössä mistä tahansa, silloin kun käyttäjä itse haluaa ja sen aikaa kuin käyttäjä itse haluaa. Verkossa olevan opetusmateriaalin päivittäminen ja muokkaaminen on erittäin helppoa ja nopeaa, eikä itse opiskelijoihin tarvitse olla suoraan yhteydessä. Multimedia tarjoaa monipuoliset tekniset mahdollisuudet toteuttaa äänen, kuvan, tekstin, videon ja animaation yhdistelmiä. Myös vuorovaikutusmahdollisuuksia, kuten keskustelufoorumeita tai chatteja voidaan tarjota käyttäjille.

Nykyisin tarjolla olevin nopeiden internetyhteyksien, suurten tallennuskapasiteettien ja monipuolisten verkko-palveluiden ansiosta on myös kuvien ja videoiden jakaminen tullut erittäin helpoksi ja nopeaksi. Voidaankin todeta, että multimedian mahdollisuudet opiskelumateriaalin tuottamisessa ja jakamisessa paranevat jatkuvasti. Nykyisin varsinaisia teknisiä rajoituksia ei enää ole, sillä hyvällä internetyhteydellä voidaan laadukastakin videota jakaa suoratoistotekniikalla, eikä käyttäjän tarvitse odottaa latautumista ollenkaan. Tilanne oli toinen vaikkapa 10 vuotta sitten.

Multimedian huonona puolena on se, ettei verkossa tarjottava opiskelu ole yleensä kovinkaan yksilöä huomioon ottavaa. Luokkatilassa opettaja voi keskittyä auttamaan hitaimmin oppivia ja opetustahtia voidaan säätää heidän mukaansa. Verkkokurssilla käytävä materiaali on monesti ennalta tehty, eikä aikataulu yleensä ota huomioon eri opiskelijoiden tarpeita.

Multimedia ei ole paras tapa opettaa esimerkiksi jonkin fyysisen laitteen tai esineen käsittelyä tai toimintaa. Kuvat, video, animaatio tai selitykset eivät pysty täysin korvaamaan sitä, että opiskelija pääsisi käyttämään laitetta omin käsin.

Multimedian parhaita puolia opetuksessa ovat sen jaettavuus verkon välityksellä, sekä se, ettei opiskelija tai opettaja ole sidottuna paikkaan tai aikaan. Multimedia on erityisen vahva opetusväline silloin, kun opetetaan jotain tietokoneeseen liittyvää tai havainnollistetaan jotain abstraktia asiaa. Opetuksen laadukkuus on kuitenkin viimekädessä kiinni materiaalin ja opetuksen laadukkuudesta, eli toisin sanoen opiskelijan, eli käyttäjän huomioimisesta.

### **Miten rakennetaan hyvä multimedia tuote?**

Hyvän multimediatuotteen suunnittelu lähtee käyttäjän tarpeista. Tekijän tulee selvittää millainen käyttäjä on, miten hän toimii ja mitä hän tarvitsee ja haluaa. Myös käyttäjän toimintaympäristö sekä hänen käyttämänsä laitteistot ja ohjelmistot tulee huomioida.

Hyvä multimediatuote on toiminnaltaan läpinäkyvä, eli sen toiminnan taustalla oleva logiikka käy ilmi käyttöliittymästä ja sovelluksen tarjoamista ohjeistuksista. Käyttäjän ei siis tarvitse arvata, mitä seuraavaksi tulisi tehdä tai mitkä hänen toimintansa seuraamukset ovat. Käyttöliittymän tulee olla käytettävä, selkeä ja johdonmukainen. Ryhmittelyä, typografiaa, värejä ja grafiikkaa yleisesti tulee miettiä. Ääntä tulee käyttää harkiten.

### **Millainen on käytettävyydeltään onnistunut multimediatuote?**

Kun potentiaalista käyttäjää pyydetään käyttämään tuotetta, hänen tulee kyetä päättämään mitä hän voi tehdä, eli mitkä ovat hänen vaihtoehtonsa ja mikä on tässä tilanteessa oikea vaihtoehtoista.

Käyttäjä ei tee turhia työvaiheita. Tämä korostuu erityisesti sellaisten sovellusten kanssa, jotka ovat esimerkiksi ammattilaisten työkäytössä. Heille tulee tarjota mahdollisimman ”nopeita” käyttöliittymiä ja erilaisia oikopolkuja, kuten näppäinoikopolkuja. Sen sijaan jos tuotteen käyttäjäkunta koostuu satunnaisemmista käyttäjistä, tulee tuotteen olla ennen kaikkea selkeä käytettävä.

Yhdeksi lauseeksi tiivistettynä tuote on käytettävyydeltään hyvä jos käyttäjä on sitä mieltä. Niinpä käyttäjälähtöisyys ja tulevien käyttäjien tekemä testaus ovat ensisijaisen tärkeitä.

### **5.3 Tulosten hyödyntäminen ja jatkotutkimus**

Käytännön työni tuloksia hyödynnetään koulun opetuksessa. Tosin kuka tahansa voi löytää videoni YouTubesta, joten niitä voi käyttää kaikki internetin käyttäjät, luonnollisesti ensisijaisesti suomen kieltä osaavat. Teoriaosuutta voivat varmasti-kin hyödyntää esimerkiksi opiskelijat, jotka tekevät jonkinlaista multimediaa koulutyönä ja haluavat ottaa selville, että miten työtä kannattaa lähestyä ja mitä tulee ottaa huomioon.

Jatkotutkimusten tarvetta olisi mielestäni lähinnä tuottamani sisällön arviointi käyttäjien suorittamien testausten ja arviointien avulla. Näin saataisiin selville, onko tässä tapauksessa opetusvideot käyttäjän mielestä toimivia ratkaisuja ja toisaalta, miten hyvin toteutus on onnistunut tämän lähestymistavan puitteissa. Myös sitä voitaisiin tutkia, kuinka hyvin multimedia sopii jonkin toisen, kenties vaikeammin multimedialla opetettavan aiheen opetukseen.

## LÄHTEET

FileInfo.com.[online]. Päivitetty 4.7.2010. [Viitattu maaliskuussa 2010] saatavilla  
www-muodossa <URL:<http://www.fileinfo.com/extension/avi>>

CVISION. [online]. [Viitattu maaliskuussa 2010] Saatavissa www-muodossa  
<URL: <http://www.cvisiontech.com/file-formats/pdf/pdf-lossy-compress.html>>

Keränen, Vesa, Lamberg, Niko & Penttinen Jukka, 2003. Digitaalinen viestintä.  
Porvoo. WS Bookwell.

Keränen, Vesa, Lamberg, Niko, Penttinen, Jukka 2006. Web-julkaiseminen &  
multimedia. Porvoo. WS Bookwell.

Ketamo, Harri & Multisilta, Jari, 2004. Multimedia. Nyt. Tampere. TTY-PAINO.

Vainionpää, Jorma 2006. Erilaiset oppijat ja oppimateriaalit verkko-opiskelussa.  
Tampere. Tampereen yliopistopaino Oy – Juvenes Print.

Välinen, Kukka-Maaria 2007. [online]. Helsinki. [Viitattu 18.5.2010] Saatavilla  
www-muodossa: <URL:  
[http://www.cs.helsinki.fi/u/kvalinen/tiedostot/Valinen\\_Kukka\\_Arvioint-  
imenetelmat\\_tutkielma.pdf](http://www.cs.helsinki.fi/u/kvalinen/tiedostot/Valinen_Kukka_Arvioint-<br/>imenetelmat_tutkielma.pdf)>

## **LIITTEET**

**Liite 1. Nielsenin tarkistuslista**

**Liite 2. Tehtävien koodi**

**Liite 1. Nielsenin tarkistuslista**

- 1 Visibility of system status
- 2 Match between system and the real world
- 3 User control and freedom
- 4 Consistency and standards
- 5 Error prevention
- 6 Recognition rather than recall
- 7 Flexibility and efficiency of use
- 8 Aesthetic and minimalist design
- 9 Help users recognize, diagnose, and recover from errors
- 10 Help and documentation

## **Liite 2. Tehtävien koodi**

### **Harjoitus 1, painike**

```
var painettuKertaa: Number = 0;

laske.addEventListener(MouseEvent.CLICK, laskeYhteen);

function laskeYhteen(event:MouseEvent):void {
    painettuKertaa++;
    teksti.text = ("Laske-painiketta on painettu " + painettuKertaa + " kertaa.");
}
```

### **Harjoitus 2, netti nappula**

```
avaaURL_btn.addEventListener(MouseEvent.CLICK, siirryURL);

function siirryURL(event:MouseEvent):void
{
    var sivusto:URLRequest = new URLRequest ("http://" + teksti.text);

    navigateToURL(sivusto);
}
```

### **Harjoitus 3, teksti ulkoisesta tiedostosta**

#### **Kehys (frame) 1.**

```
stop();

tuoTeksti.addEventListener(MouseEvent.CLICK, mene);

function mene(event:MouseEvent):void {

    gotoAndStop(2);
}
```

#### **Kehys 2.**

```
var lataaja:URLLoader = new URLLoader(new URLRequest("teksti.txt"));
lataaja.addEventListener(Event.COMPLETE, valmis);

function valmis(event:Event):void {

    var ladattuTeksti:URLLoader = URLLoader(event.target);
    tekstiKenttä.text = ladattuTeksti.data;
}

poistaTeksti.addEventListener(MouseEvent.CLICK, palaa);
```



```
function palaa(Event:MouseEvent):void {

    gotoAndStop(1);
}
```

#### **Harjoitus 4, laskin**

```
plus.addEventListener(MouseEvent.CLICK, plusClick);
miinus.addEventListener(MouseEvent.CLICK, miinusClick);
kertaa.addEventListener(MouseEvent.CLICK, kertaaClick);
jaettuna.addEventListener(MouseEvent.CLICK, jaettunaClick);
yhtäKuin.addEventListener(MouseEvent.CLICK, yhtäKuinClick);
```

```
var plusMerkki:Boolean = false;
var miinusMerkki:Boolean = false;
var kertaaMerkki:Boolean = false;
var jaettunaMerkki:Boolean = false;
```

```
input1.border = true;
input2.border = true;
input1.restrict = "0-9";
input2.restrict = "0-9";
```

```
function plusClick(event:MouseEvent): void{

    plusMerkki = true;
    miinusMerkki = false;
    kertaaMerkki = false;
    jaettunaMerkki = false;
    merkki_teksti.text = "+";

}
```

```
function miinusClick(event:MouseEvent): void{

    plusMerkki = false;
    miinusMerkki = true;
    kertaaMerkki = false;
    jaettunaMerkki = false;
    merkki_teksti.text = "-";

}
```

```
function kertaaClick(event:MouseEvent): void{

    plusMerkki = false;
    miinusMerkki = false;
    kertaaMerkki = true;
```

```

    jaettunaMerkki = false;
    merkki_teksti.text = "x";

}

function jaettunaClick(event:MouseEvent): void{

    plusMerkki = false;
    miinusMerkki = false;
    kertaaMerkki = false;
    jaettunaMerkki = true;
    merkki_teksti.text = "/";

}

var arvo1:String;
var arvo2:String;
var plusTulos:Number
var miinusTulos:Number
var jaettunaTulos:Number;
var kertaaTulos:Number;

function yhtäKuinClick(event:MouseEvent):void {
    arvo1=input1.text
    arvo2=input2.text

    if(plusMerkki == true) {

        plusTulos = parseInt(arvo1) + parseInt(arvo2);
        plusTulos.toString();
        tulos.text = String(plusTulos);

    } else if(miinusMerkki == true) {
        miinusTulos = parseInt(arvo1) - parseInt(arvo2);
        miinusTulos.toString();
        tulos.text = String(miinusTulos);

    } else if(kertaaMerkki == true) {
        kertaaTulos = parseInt(arvo1) * parseInt(arvo2);
        kertaaTulos.toString();
        tulos.text = String(kertaaTulos);

    } else if(jaettunaMerkki == true) {
        jaettunaTulos = parseInt(arvo1) / parseInt(arvo2);
        jaettunaTulos.toString();
        tulos.text = String(jaettunaTulos);

    }
}

```

```
}
```

### **Harjoitus 5, digitaalinen kello**

```
var PäiväMäärä = new Date();
var sekunnit = PäiväMäärä.getSeconds();
var minuutit = PäiväMäärä.getMinutes();
var tunnit = PäiväMäärä.getHours();
var päivä = PäiväMäärä.getDay();
var PVM = PäiväMäärä.getDate();
var kuukausi = PäiväMäärä.getMonth();
var vuosi = PäiväMäärä.getFullYear();
```

```
if (tunnit == 0) {
    tunnit = 12;
}
if (minuutit < 10) {
    minuutit = "0" + minuutit;
}
if (sekunnit < 10) {
    sekunnit = "0" + sekunnit;
}
```

```
if (päivä == 0) {
    päivä = "sunnuntai"
} else if (päivä == 1) {
    päivä = "maanantai"
} else if (päivä == 2) {
    päivä = "tiistai"
} else if (päivä == 3) {
    päivä = "keskiviikko"
} else if (päivä == 4) {
    päivä = "torstai"
} else if (päivä == 5) {
    päivä = "perjantai"
} else if (päivä == 6) {
    päivä = "lauantai"
}
```

```
if (kuukausi == 0) {
    kuukausi = "tammikuuta"
} else if (kuukausi == 1) {
    kuukausi = "helmikuuta"
} else if (kuukausi == 2) {
    kuukausi = "maaliskuuta"
} else if (kuukausi == 3) {
```

```

        kuukausi = "huhtikuuta"
    } else if (kuukausi==4){
        kuukausi = "toukokuuta"
    } else if (kuukausi==5){
        kuukausi = "kesäkuuta"
    } else if (kuukausi==6){
        kuukausi = "heinäkuuta"
    } else if (kuukausi==7){
        kuukausi = "elokuuta"
    } else if (kuukausi==8){
        kuukausi = "syyskuuta"
    } else if (kuukausi==9){
        kuukausi = "lokakuuta"
    } else if (kuukausi==10){
        kuukausi = "marraskuuta"
    } else if (kuukausi==11){
        kuukausi = "joulukuuta"
    }

```

```

aika.text = ((tunnit) + ":" + (minuutit) + ":" + (sekunnit));
päivämäärä.text = ((päivä) + " " + (PVM) + " " + (kuukausi) + " " + (vuosi));

```

## **Harjoitus 6, Flash sovellus Flash sovelluksessa**

### **Kehys 1.**

```

stop();
var xsijainti:Number = 0;
var ysijainti:Number = 0;
var xsijainti2:Number = 20;
var ysijainti2:Number = 100;
var kellonLataaja:Loader = new Loader();
var laskimenLataaja:Loader = new Loader();

var kello:URLRequest = new URLRequest("kello.swf");

kellonLataaja.load(kello);
kellonLataaja.x = xsijainti;
kellonLataaja.y = ysijainti;
addChild(kellonLataaja);

avaa.addEventListener(MouseEvent.CLICK, avaaLaskin);

function avaaLaskin(event:MouseEvent):void {

    var laskin:URLRequest = new URLRequest("Laskin.swf");
    laskimenLataaja.load(laskin);
    laskimenLataaja.x = xsijainti2;
    laskimenLataaja.y = ysijainti2;

```

```

        addChild(laskimenLataaja);
        gotoAndStop(2);
    }

```

## **Kehys 2.**

```

sulje.addEventListener(MouseEvent.CLICK, suljetaan);

```

```

function suljetaan(event:MouseEvent):void {

    removeChild(laskimenLataaja);
    gotoAndStop(1);

}

```

## **Harjoitus 7, galleria**

```

var galleriaKansio:String = "Galleria_kuvat";

```

```

var kuva1lataaja:Loader = new Loader();
var kuva2lataaja:Loader = new Loader();
var kuva3lataaja:Loader = new Loader();
var kuva4lataaja:Loader = new Loader();
var kuva5lataaja:Loader = new Loader();

```

```

var pikku_kuva1haku:URLRequest = new URLRequest(galleriaKansio + "/Pienet/1_pieni.jpg");
var pikku_kuva2haku:URLRequest = new URLRequest(galleriaKansio + "/Pienet/2_pieni.jpg");
var pikku_kuva3haku:URLRequest = new URLRequest(galleriaKansio + "/Pienet/3_pieni.jpg");
var pikku_kuva4haku:URLRequest = new URLRequest(galleriaKansio + "/Pienet/4_pieni.jpg");
var pikku_kuva5haku:URLRequest = new URLRequest(galleriaKansio + "/Pienet/5_pieni.jpg");

```

```

kuva1lataaja.load(pikku_kuva1haku);
kuva2lataaja.load(pikku_kuva2haku);
kuva3lataaja.load(pikku_kuva3haku);
kuva4lataaja.load(pikku_kuva4haku);
kuva5lataaja.load(pikku_kuva5haku);

```

```

pikku_kuva1lataaja.addChild(kuva1lataaja);
pikku_kuva2lataaja.addChild(kuva2lataaja);
pikku_kuva3lataaja.addChild(kuva3lataaja);
pikku_kuva4lataaja.addChild(kuva4lataaja);
pikku_kuva5lataaja.addChild(kuva5lataaja);

```

```

pikku_kuva1lataaja.buttonMode = true;
pikku_kuva2lataaja.buttonMode = true;
pikku_kuva3lataaja.buttonMode = true;
pikku_kuva4lataaja.buttonMode = true;
pikku_kuva5lataaja.buttonMode = true;

```

```

pikku_kuva1lataaja.addEventListener(MouseEvent.CLICK, pikku_kuva1click);
pikku_kuva2lataaja.addEventListener(MouseEvent.CLICK, pikku_kuva2click);
pikku_kuva3lataaja.addEventListener(MouseEvent.CLICK, pikku_kuva3click);
pikku_kuva4lataaja.addEventListener(MouseEvent.CLICK, pikku_kuva4click);
pikku_kuva5lataaja.addEventListener(MouseEvent.CLICK, pikku_kuva5click);

```

```

var kuva1haku:URLRequest = new URLRequest(galleriaKansio + "/1.jpg");
var kuva2haku:URLRequest = new URLRequest(galleriaKansio + "/2.jpg");
var kuva3haku:URLRequest = new URLRequest(galleriaKansio + "/3.jpg");
var kuva4haku:URLRequest = new URLRequest(galleriaKansio + "/4.jpg");
var kuva5haku:URLRequest = new URLRequest(galleriaKansio + "/5.jpg");

```

```

var isoKuva:Loader = new Loader();

```

```

isoKuva.load(kuva1haku);
kuva_kenttä.addChild(isoKuva);

```

```

function pikku_kuva1click(event:MouseEvent):void {

    isoKuva.load(kuva1haku)
    kuva_kenttä.removeChildAt(1);
    kuva_kenttä.addChild(isoKuva);
}

```

```

function pikku_kuva2click(event:MouseEvent):void {
    isoKuva.load(kuva2haku);
    kuva_kenttä.removeChildAt(1);
    kuva_kenttä.addChild(isoKuva);
}

```

```

function pikku_kuva3click(event:MouseEvent):void {
    isoKuva.load(kuva3haku);
    kuva_kenttä.removeChildAt(1);
    kuva_kenttä.addChild(isoKuva);
}

```

```

function pikku_kuva4click(event:MouseEvent):void {
    isoKuva.load(kuva4haku);
    kuva_kenttä.removeChildAt(1);
    kuva_kenttä.addChild(isoKuva);
}

```

```

function pikku_kuva5click(event:MouseEvent):void {
    isoKuva.load(kuva5haku);
    kuva_kenttä.removeChildAt(1);
    kuva_kenttä.addChild(isoKuva);
}

```

**Harjoitus 8, preloader**

```

this.addEventListener(Event.ENTER_FRAME, lataa);

function lataa(event:Event):void {

    var kokoTiedosto:Number = this.stage.loaderInfo.bytesTotal;
    var ladattu:Number = this.stage.loaderInfo.bytesLoaded;

    latausPalkki.scaleX = ladattu/kokoTiedosto;
    lataajaTeksti.text = Math.floor((ladattu/kokoTiedosto)*100)+ "%";

    if (kokoTiedosto == ladattu) {
        play();
        this.removeEventListener(Event.ENTER_FRAME, lataa);
    }

}

```

**Harjoitus 9, palapeli**

```

var counter:Number = 0;
var aloitaX:Number;
var aloitaY:Number;

vasenYlä.addEventListener(MouseEvent.CLICK, nosta);
vasenYlä.addEventListener(MouseEvent.CLICK, laske);

vasenAla.addEventListener(MouseEvent.CLICK, nosta);
vasenAla.addEventListener(MouseEvent.CLICK, laske);

oikeaYlä.addEventListener(MouseEvent.CLICK, nosta);
oikeaYlä.addEventListener(MouseEvent.CLICK, laske);

oikeaAla.addEventListener(MouseEvent.CLICK, nosta);
oikeaAla.addEventListener(MouseEvent.CLICK, laske);

function nosta(event:MouseEvent):void {

    event.target.startDrag(true);
    event.target.parent.addChild(event.target);
    aloitaX = event.target.x;
    aloitaY = event.target.y;
}

function laske(event:MouseEvent): void {
    event.target.stopDrag();
    var kohteenNimi:String = event.target.name + "Kohde";
    var kohde:DisplayObject = getChildByName(kohteenNimi);
}

```

```

        if (event.target.dropTarget != null && event.target.dropTarget.parent == kohde){

            event.target.removeEventListener(MouseEvent.MOUSE_DOWN, nosta);
            event.target.removeEventListener(MouseEvent.MOUSE_UP, laske);
            event.target.buttonMode = false;
            event.target.x = kohde.x;
            event.target.y = kohde.y;
            counter++;

        }else{
            event.target.x = aloitaX;
            event.target.y = aloitaY;

        }

        if(counter == 4){
            onnittelut.text = "Onneksi olkoon!";
        }
    }

```

```

vasenYlä.buttonMode = true;
vasenAla.buttonMode = true;
oikeaYlä.buttonMode = true;
oikeaAla.buttonMode = true;

```

### **Harjoitus 10, MP3-soitin**

```

var musiikki:Sound = new Sound(new URLRequest("Devini_Project_1.mp3"));
var äänikanava:SoundChannel;
var soi:Boolean = false;

```

```

stop_btn.addEventListener(MouseEvent.CLICK, pysäytäBiisi);

```

```

function pysäytäBiisi(event:Event):void {

```

```

    if(äänikanava != null)
    {
        äänikanava.stop();
        soi = false;
    }
}

```

```

play_btn.addEventListener(MouseEvent.CLICK, soitaBiisi);

```

```

function soitaBiisi(event:Event):void {

```

```

    if(!soi)

```



```

    {
        äänikanava = musiikki.play();
        soi = true;
        ääni_mc.slaideri.x = 33;

        var vol:Number = ääni_mc.slaideri.x / 100;
        var st:SoundTransform = new SoundTransform(vol);

        if (äänikanava != null)
        {
            äänikanava.soundTransform = st;
        }
    }
}

var raahaus:Boolean = false;
var viiva:Rectangle = new Rectangle(0,0,100,0);
ääni_mc.slaideri.addEventListener(MouseEvent.CLICK, raahaa);

stage.addEventListener(MouseEvent.CLICK, irrota);

function raahaa(event:Event):void {

    ääni_mc.slaideri.startDrag(false,viiva);
    raahaus = true;
    ääni_mc.slaideri.addEventListener(Event.ENTER_FRAME, säädäÄäni);
}

function irrota(event:Event):void {

    if (raahaus)
    {
        ääni_mc.slaideri.stopDrag();
        raahaus = false;
    }
}

function säädäÄäni(event:Event):void {

    var vol:Number = ääni_mc.slaideri.x / 100;
    var st:SoundTransform = new SoundTransform(vol);

    if (äänikanava != null)
    {
        äänikanava.soundTransform = st;
    }
}

```

## **Harjoitus 11, Pertin pelit**

### **Kehys 1**

```
var määrä:Number = 0;
```

### **Kehys 2**

```
keltainen.addEventListener(MouseEvent.CLICK, osuma)
```

```
var virheet:Number = 0;
```

```
if(määrä == 0) {näytä.text = "0"}
```

```
function osuma(event:MouseEvent):void {
```

```
    if(määrä > 0) {
        määrä++;
        määrä.toString();
        näytä.text = String(määrä);
    }else if(määrä == 0) {
        määrä = 1
        määrä.toString()
        näytä.text = String(määrä);
    }
}
```

```
punainen.addEventListener(MouseEvent.CLICK, painettu)
```

```
vihreä.addEventListener(MouseEvent.CLICK, painettu)
```

```
sininen.addEventListener(MouseEvent.CLICK, painettu)
```

```
function painettu (event:MouseEvent):void {
```

```
    if(virheet < 4) {
        virheet++;
        virheet.toString();
        virhe.text = String(virheet + " / 5");
    }else if(virheet == 4) {
        gotoAndStop(2);
        virhe.text = "5 / 5"
        tuloksesi.text = "Tuloksesi on";
        loppu.text = "Game Over!";
    }
}
```

### **Kehys 3**

```
var satunnaisLuku:Number = Math.floor(Math.random()*3)+1;
```

```
if(satunnaisLuku == 1) {
```

```

        gotoAndPlay(5)
    } else if (satunnaisLuku == 2) {
        gotoAndPlay(11)
    } else if (satunnaisLuku == 3) {
        gotoAndPlay(16)
    }

```

### **Kehys 9**

```
gotoAndPlay(3)
```

### **Kehys 15**

```
gotoAndPlay(3)
```

### **Kehys 21**

```
gotoAndPlay(3)
```

### **Harjoitus 12, tasohyppely**

```

package //
{
    import flash.display.MovieClip
    import flash.events.Event
    import flash.events.KeyboardEvent

    public class UlkoinenKoodi extends MovieClip
    {
        private var nopeusX:Number;
        private var nopeusY:Number;

        public function UlkoinenKoodi():void
        {
            nopeusX = 0;
            nopeusY = 0;
            alkuPiste.visible = false;

            stage.focus = stage;

            this.addEventListener(Event.ENTER_FRAME, frameenTulo);
            stage.addEventListener(KeyboardEvent.KEY_DOWN, painikePainetaan);
            stage.addEventListener(KeyboardEvent.KEY_UP, painikePäästetään);
        }
        private function frameenTulo(e:Event):void
        {
            nopeusY += 2;

            pelaaja.x += nopeusX;

```

```

        pelaaja.y += nopeusY

        käsitteleTörmäys();

        näkymäSiirtyy();
    }

private function painikePainetaan(e:KeyboardEvent):void
{
    switch (e.keyCode)
    {
        case 37:
            nopeusX = -7;
            break;

        case 38:
            nopeusY = -20;
            break;

        case 39:
            nopeusX = 7;
            break;

        default:

    }
}

private function painikePäätetään(e:KeyboardEvent):void
{
    switch (e.keyCode)
    {
        case 37:
        case 39:
            nopeusX = 0;
            break;

        default:

    }
}

private function käsitteleTörmäys():void
{
    if(nopeusY > 0)
    {
        if (pelaaja.y > stage.stageHeight)
        {
            pelaaja.x = alkuPiste.x;
            pelaaja.y = alkuPiste.y;
            rajat.x = 0;
            rajat.y = 0;
        }
    }
}

```

```

        nopeusY = 0;
    } else {
        var törmäys:Boolean = false;

        if(rajat.hitTestPoint(pelaaja.x, pelaaja.y, true))
        {
            törmäys = true;
        }
        if(törmäys)
        {
            while(törmäys)
            {
                pelaaja.y -= 0.1;
                törmäys = false;

                if (rajat.hitTestPoint(pelaaja.x, pelaaja.y, true))
                {
                    törmäys = true;
                }
            }
            nopeusY = 0;
        }
    }
}

private function näkymäSiirtyy():void
{
    rajat.x += (stage.stageWidth * 0.5) - pelaaja.x;
    pelaaja.x = stage.stageWidth * 0.5;
}
}

```